In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

7/25/68

DESIGN OF A MECHANICAL PHASE PLANE TIME RESPONSE ANALYZER

A THESIS

Presented to

The Faculty of the Graduate Division

by

Charles Richard Scraggs

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mechanical Engineering

Georgia Institute of Technology

June, 1970

DESIGN OF A MECHANICAL PHASE PLANE

TIME RESPONSE ANALYZER

Approved:

Chairman

Date approved by Chairman: 5/11/70

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

Many physical systems are best described by second order, nonlinear, differential equations. Examples of such systems abound in almost every aspect of engineering and scientific analysis. In particular, the fields of control engineering, mathematics, mechanical vibrations, dynamics of machinery, and electrical engineering draw many of their standard or typical function equations from second order descriptions.

Unfortunately, differential equations with nonlinearities usually are difficult or impossible to solve exactly. Analog or digital computers yield accurate solutions or approximations, but this approach is usually time consuming, especially when it is desired to glean only particular pieces of information from the complete solution. In this case, the phase plane technique can yeild a considerable amount of very useful information through easily applied graphical techniques. The analysis may be carried as far as desired, but often a simple sketch of a phase trajectory, far short of a complete solution will be sufficient to allow the analyst or designer to draw conclusions about the response of the system.

To the design engineer, time response information is especially useful. With phase plane analysis, many aspects of the transient response such as settling time, overshoot, rise time, limit cycles, and others are readily determined or measured. Once the phase "portrait" or general mapping of the system is obtained, the designer may vary such parameters

as mass, damping, or forcing to alter or improve his design.

The coordinates normally used on the phase plane are displacement or position, x, and velocity, $\dot{x}$. It can be shown that the independent variable time can be approximated from these two dependent variables. The customary procedure in time response analysis is to follow one of three methods, each graphical in nature, to approximate increments of time between various system positions or states. For a given equation describing a system, there exists a unique phase portrait. A particular trajectory of this portrait will be determined by the initial conditions peculiar to it — different initial conditions resulting in different trajectories. However, once the portrait is obtained, any trajectory may be quickly and accurately mapped. On the other hand, the graphics of obtaining a time response must be laboriously performed on each trajectory.

The objective of this thesis then is to design a mechanical device that can accept as inputs x and $\dot{x}$ from a general phase plane mapping, and by functionally operating on these two inputs, compute the elapsed time from point to point. At any position on the trajectory, the device, henceforth referred to as the computer, will give the values of x, $\dot{x}$, and the approximated elapsed time from the initial conditions of the system to the attainment of the system state at that time. Such a computer would allow rapid tracings of many trajectories of a single portrait. Functionally it would be of desk top size and readily usable on any flat table surface. Ideally it would increase the appeal of phase plane time response analysis by omitting further laborious graphical procedures once the system trajectory is obtained.

CHAPTER II

THEORY AND FUNDAMENTALS OF MECHANIZATION

Methods of Operation

Since $\dot{x}$ and x are the corresponding ordinate and abscissa of the phase portrait, the following relationships apply:

$$dx/dt = \dot{x}$$

$$dx = \dot{x}dt$$

$$x = \int_{o}^{t} \dot{x}dt$$

$$\Delta x = \int_{t1}^{t2} \dot{x}dt$$

$$\Delta x = \dot{x}_{avg} \Delta t$$

$$\Delta t = \Delta x/\dot{x}_{avg} \qquad (2\text{-}1)$$

Equation (2-1) is the basic equation for approximating an increment of time, Δt. Three methods will be discussed. The first necessitates plotting $|1/\dot{x}|$ vs $|x|$ to give a new mapping, the area under this curve being proportional to time. The second method uses a technique of dividing the trajectory into circular arc segments whose centers lie on the x - axis. Time is then proportional to the angle subtended by these arcs. The third method approximates the trajectory with straight line segments whose projections on the x - axis are Δx increments. The center point of each segment is $\dot{x}_{avg}$ and the incremental time is the ratio of Δx to $\dot{x}_{avg}$ (1).

Any mechanical device to increment time will require x and $\dot{x}$ to be input in some form, whether it be a direct or trigonometric proportioning of these variables. The first method of replotting the trajectory into a $|1/\dot{x}|$ vs $|x|$ mapping, besides compounding the labor of the graphics, has limiting drawbacks for mechanization. It is necessary to realize that both $\dot{x}$ and x oscillate through zero values as the trajectory is traced on the general phase portrait. Therefore, since $\dot{x}$ appears in the denominator of the approximating equation, the result at iterations where $\dot{x}$ approaches zero becomes unbounded. No mechanism will follow such a condition. Where approximations are attempted in such cases, linkages or movements jam or realistically become preponderous in size to correspond to an unbounded state. For example, the $|1/\dot{x}|$ vs $|x|$ mapping would seem an attractive scheme since actually $\dot{x}$ and x could be directly received as displacements from the original portrait. The reciprocal of $\dot{x}$ could be formed by the computer, and through a series of three integrations with feedback inputs not unlike an ordinary electric analog computer, the area under the reciprocal curve would be integrated with respect to x and the resultant elapsed time obtained. Mechanical ball and disk integrators could make up the basic computer. Figure 1 shows schematically how such a device might be configured. The $\dot{x}$ input and the reciprocal integrator are used to form $1/\dot{x}$. The reciprocal must be powered externally and the $\dot{x}$ input is used to bias the $|1/\dot{x}|$ output much like a vacuum tube is controlled by the signal voltage. It is easily seen that as $\dot{x}$ approaches zero, the x input integrator advances

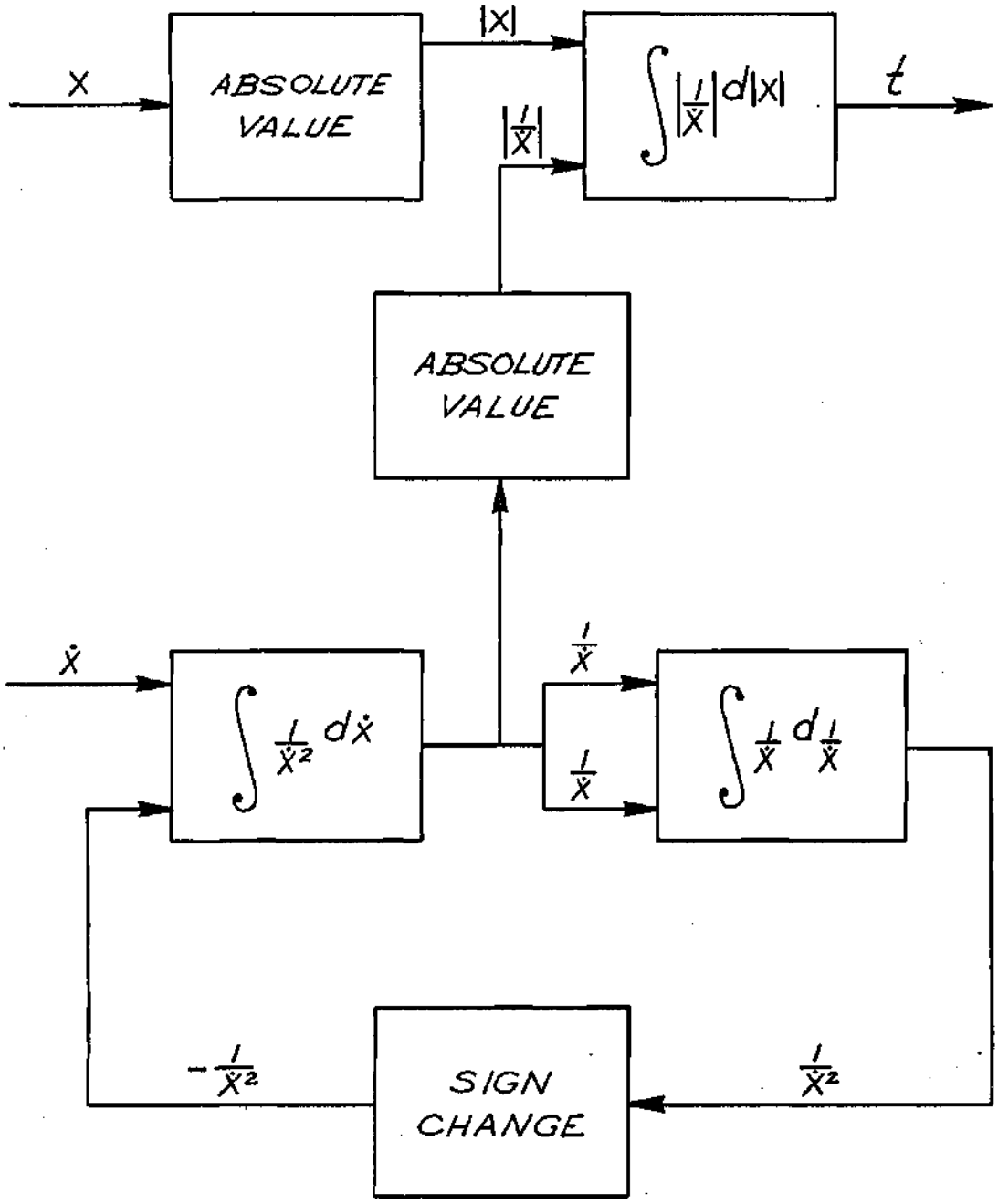


Figure 1. Block Diagram of $|1/\dot{x}|$ vs $|x|$ Integration.

CRS

to an unbounded value of time. The mechanism could be designed to "saturate" and record all unbounded values of time as a single large increment, but this is not a desirable feature.

In mechanizing a scheme for circular arc approximations, the approach shown in Figure 2 could be used. From the calculus and trigonometry

$$dx = \dot{x}dt$$

$$\dot{x} = (1)\sin\theta$$

$$dx = (1)\sin\theta \; d\theta$$

$$dt = dx/\dot{x} = d\theta \tag{2-2}$$

From polar coordinates the following is true:

$$(1)\sin\theta = (r)\sin\alpha \tag{2-3}$$

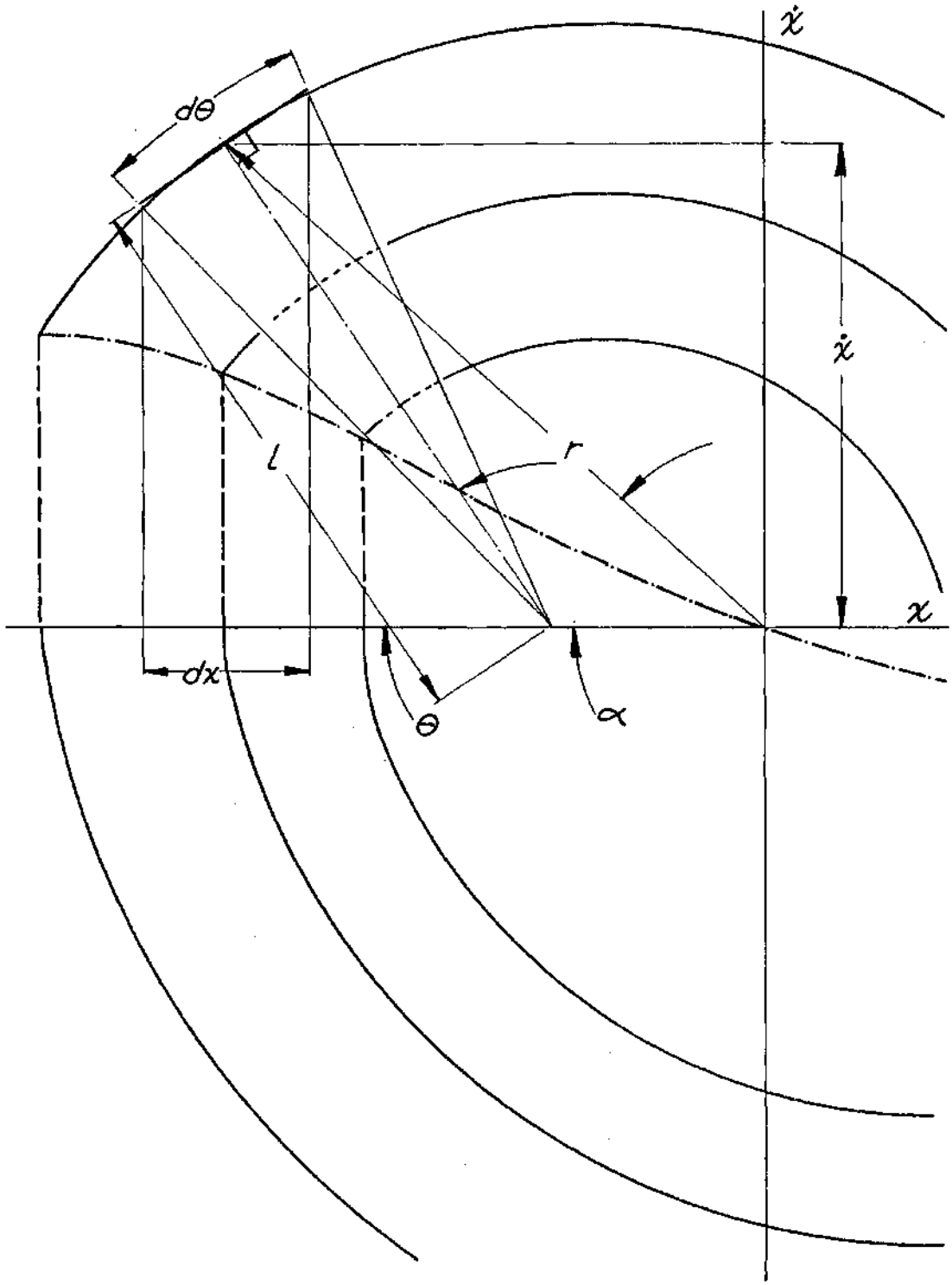
$$x + (1)\cos\theta = (r)\cos\alpha \tag{2-4}$$

Differentiating (2-4) and noting that for a circular arc, the center at x,

$$(\dot{1}) = \dot{x} = o$$

$$-(1)\sin\theta\dot{\theta} = (\dot{r})\cos\alpha - (r)\sin\alpha\,\dot{\alpha}$$

Now the following relationships result:



CRS

Figure 2. Method of Circular Arc Approximation.

$$(1)\sin\theta = (r)\sin\alpha$$

$$-(r)\sin\alpha\dot{\theta} = (\dot{r})\cos\alpha - (r)\sin\alpha\dot{\alpha}$$

$$\dot{\theta} = -(\dot{r}/r)\cot\alpha + \dot{\alpha}$$

$$d\theta/dt = 1/dt\left[-(dr/r)\cot\alpha + d\alpha\right]$$

$$d\theta = dt = -(dr/r)\cot\alpha + d\alpha \qquad (2\text{-}5)$$

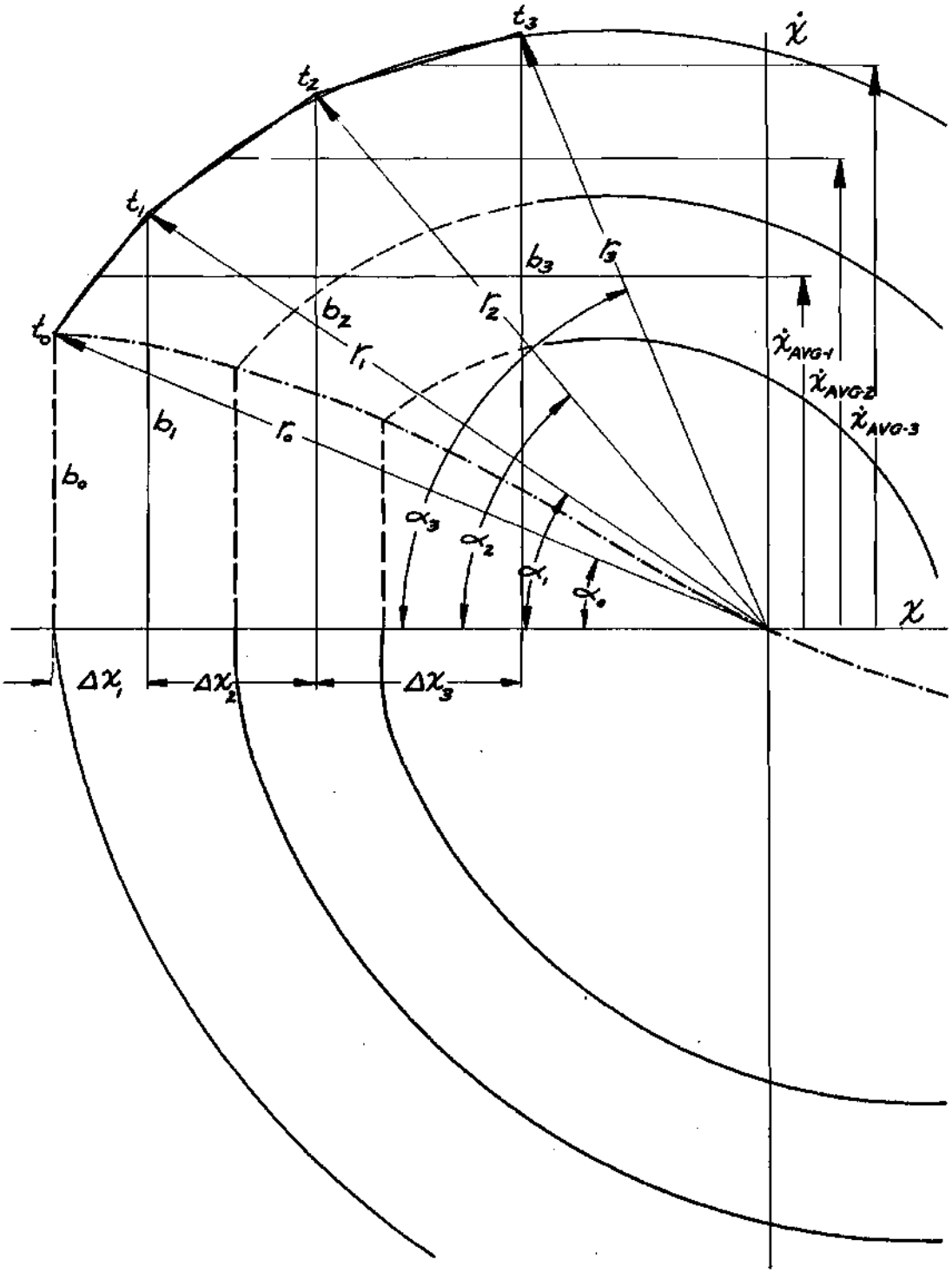
Equation (2-5) relates that for a circular trajectory the differential, dr, vanishes and the elapsed time is proportional to the angular displacement. Note however, as $\dot{x}$ oscillates through zero, the value of the cotangent term becomes unbounded. This characteristic of the defining equation also makes this method unattractive by reason of mechanical problems already stated.

The third method using Equation (2-1) allows a satisfactory design since the problem of an unbounded variable is circumvented. The average value of the velocity need never be zero, and in an actual mechanism, provision can be made so that the minimum magnitude of $\dot{x}$ can be controlled. The method by which Equation (2-1) is synthesized is illustrated in the subsequent analysis.

From Figure 3, the average values of three approximating segments of a typical trajectory are shown. For example, the average value of the trapezoid partitioned by b_1 and b_2 is

$$\frac{b_2 + b_1}{2} = \dot{x}_{avg\ 2}$$

In general, it is obvious that the average value of the velocity is



CRS

Figure 3. Method of Δx Partitioning Approximation.

given by

$$\frac{b_n + b_{n-1}}{2} = \dot{x}_{avg-n} \qquad n = 1, 2 \ldots k$$

Note that if r_n is the polar vector length from the origin to the Nth increment on the trajectory, and α_n is the polar angle measured from the x - axis, the vertical and horizontal partition projections are given by

$$b_n = r_n \sin \alpha_n$$

$$|\Delta x_n| = |(r_{n-1} \cos \alpha_{n-1}) - (r_n \cos \alpha_n)|$$

To compute the elapsed time over a number of Δx increments of displacements, it is necessary to form all $\dot{x}_{avg-n}$ values, divide the Δx_n values by these, and sum the resulting increments of time Δt_n. All these operations can be performed mechanically.

Functional Operation of the Computer

From Figure 3 it is seen that the actual inputs of such a computer are the polar variables r_n and α_n and $\dot{x}$ and x are $(r_n \sin \alpha_n)$ and $(r_n \cos \alpha_n)$ respectively. Using these two inputs as continuous variables, the output, elapsed time, is computed from segment to segment of the approximating trajectory. A block diagram analysis of the computer operation is shown in Figure 4. The "Sine-Cosine Function Multipliers" mechanically construct the $\dot{x}$ and x variables. The "Iterative Adder"

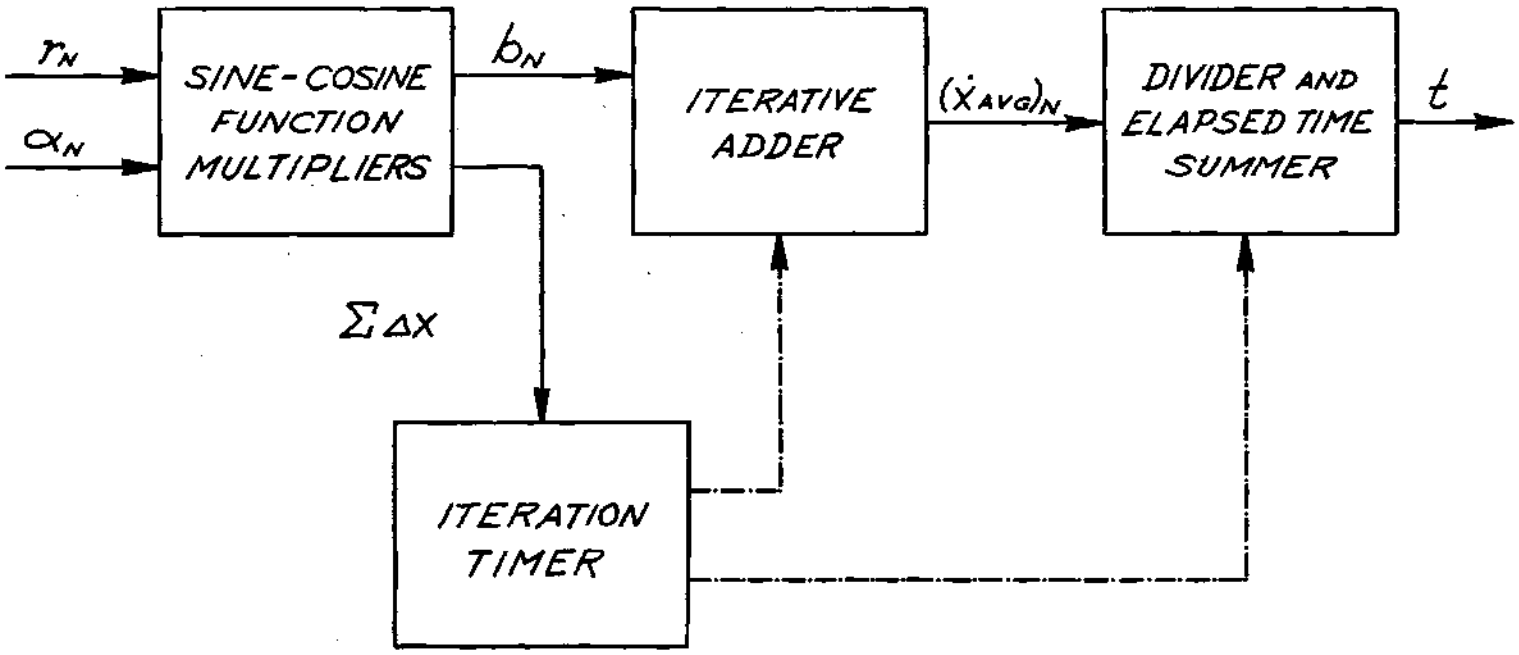


Figure 4. Block Diagram of Computer Operation.

CRS

accepts as inputs ($r_n \sin \alpha_n$) and a sequenced operational signal from the "Iteration Timer." This timer functions to make the mechanisms digital in nature. Consider now that all Δx projections in Figure 3 are equal. Each time a Δx segment is passed as the polar vector sweeps around the trajectory, the timer causes a $\dot{x}_{avg-n}$ value and a corresponding Δt increment to be formed and summed. The actual dwell period within each segment allows for a storage of a Δt_n and (b_{n-1}) value and relaxation of the related mechanisms to ready the computing action for the next iteration.

The objective of any such design must express several basic concepts. The accuracy of the computer should match or exceed that of the presently used graphical techniques. However, error analysis should consider the graphical inaccuracies of the general sketched trajectory itself. Ease and flexibility of operation are important design factors. Some provision should be included to change scale factors so that all general trajectories might be analyzed. Economy of manufacturing is always a major consideration. Finally, the aesthetic quality of the device is a consideration of all commercial design that can preclude its full utility by the user.

CHAPTER III

DESIGN OF THE COMPUTER

There are usually alternatives in any design and choices are made in accordance with the final objectives. A device such as the proposed computer necessitates the supposition of limiting measures or features early in the design process. It is expedient to design the overall device as a system of components producing the desired end result or output. This is the method of attack that will be followed in the design of the phase plane time response analyzer.

Mechanism Considerations

Some limitations should be decided upon so far as the allowable size of the phase portrait. Too, the displacement increment, Δx, is chosen a constant value to simplify the division operation. A judicious choice must be made since too large a value will generally increase error in the approximation, while too small a value will cause high speed iteration cycles and resultant dynamic problems in the mechanisms. Some scale factor is also necessary in such mechanisms as multipliers. However, it is obvious that as the scale factor, which would be the mechanism to portrait size ratio, becomes very small, the accuracy and precision of the mechanisms must be likewise increased so that errors here do not contribute substantially to output error percentage wise. If this ratio is large, the mechanism is large and the device loses its "desk top" appeal.

The iteration timer is a device to initiate compute action at the beginning, during, and at the end of each cycle. A cycle here always refers to a Δx displacement and each Δx increment traversed along the abscissa causes a Δt to be produced. The iterative adder must store the preceding value of (b_n) sequentially at each iteration to be used in computation of a particular Δt value. These physical displacements may be stored as potential energy. To minimize inaccuracies caused by clearances, backlash, or the like, gearing will be avoided when possible.

Physical Limits

For the final design, the limits of the phase portrait are as shown in Figure 5. However, it is important to notice that the most general portrait can be sketched within these limits by a simple change of scale. As shown, the trajectory must fit within a circle of radius six inches to be within the range of the computer. Now consider a general trajectory where initial conditions are $\dot{x} = o$ and the displacement is an arbitrary value. The minimum angle at which the trajectory emanates from or converges to the abscissa will present a limiting maximum value of Δt. The ratio of Δx to $\dot{x}_{avg}$ can be chosen equal to an arbitrary value Q. From consideration of all aspects of the design, this Q value was chosen equal to two. As Figure 5 shows, this allows an angle of incidence or departure of 45^{o} as a limiting minimum. This in itself will handle the majority of the trajectories encountered since the curve must always cross the x - axis with an infinite slope. This makes the trajectory at crossover nearly vertical for some distance

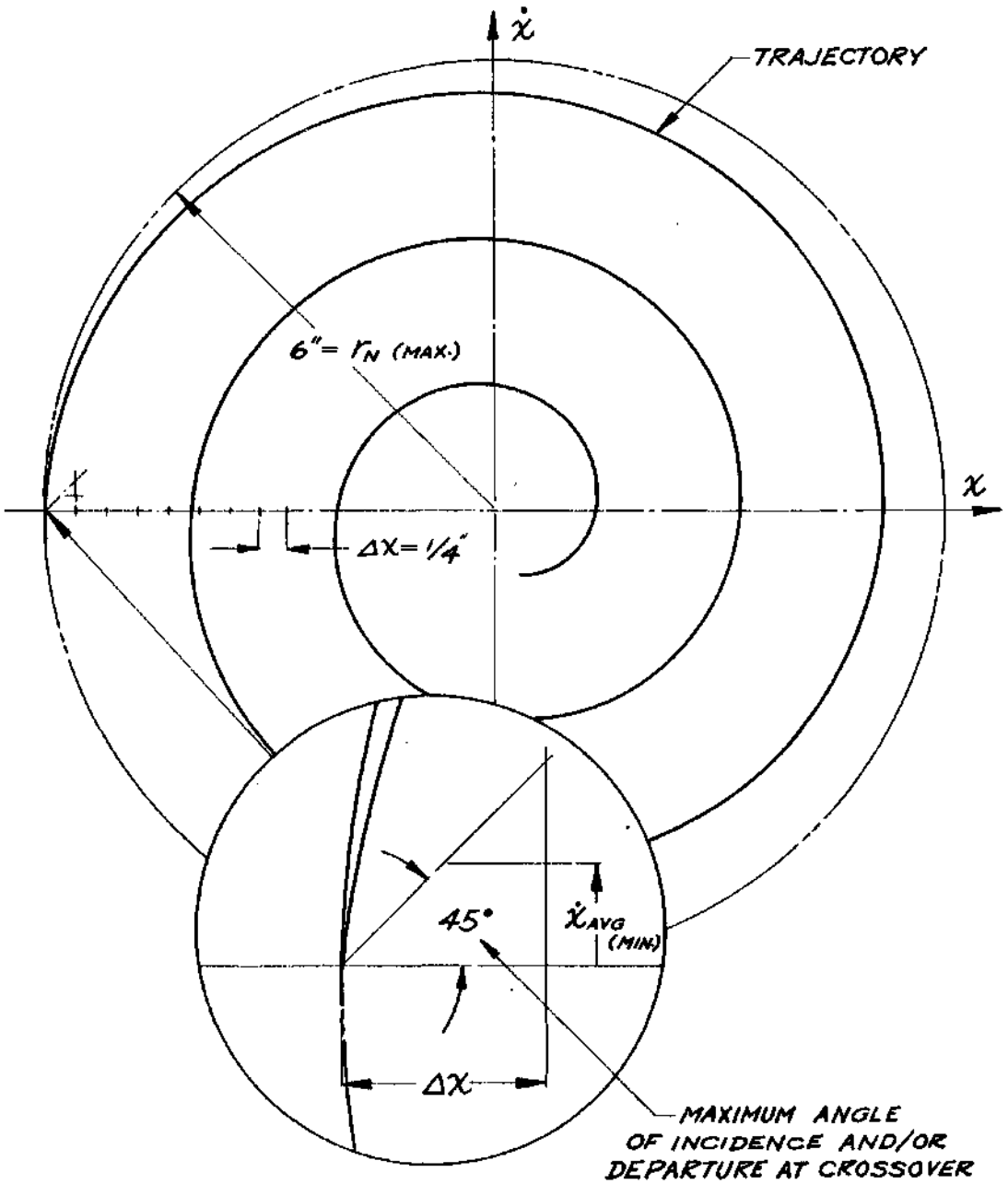


Figure 5. Physical Limits of Phase Portrait.

CRS

on both sides of the coordinate axis. If exceptions arise, a change of scale on the ordinate or abscissa will alter the curve such that the crossover angle will be suitable.

Finally Δx was chosen constant and equal to one-fourth inch. Again, a scale factor can be employed where this increment size seems unappropriate.

Component Mechanisms

The computer, considered as a system, is composed of a number of components. The remainder of this chapter will be spent in a complete exposition of each component and its final integration into the operation of the computer. The following listing is in order of input to output sequence as the mechanism components combine to make up the unitized computing device.

1. Input Assembly
2. Sine-Cosine Function Multipliers
3. Iteration Timer and Iterative Adder
4. Divider and Elapsed Time Summer

Input Assembly

The function of the input assembly is to receive from the phase trajectory the input variables (r_n) and (α_n) through a tracing stylus and admit these values to the computing mechanisms. Referring to Figure 6, Stylus Input Assembly, the vertical axis of the input spindle is centered over the origin of the phase portrait. The mechanical functioning of all the mechanisms is designed to operate from the

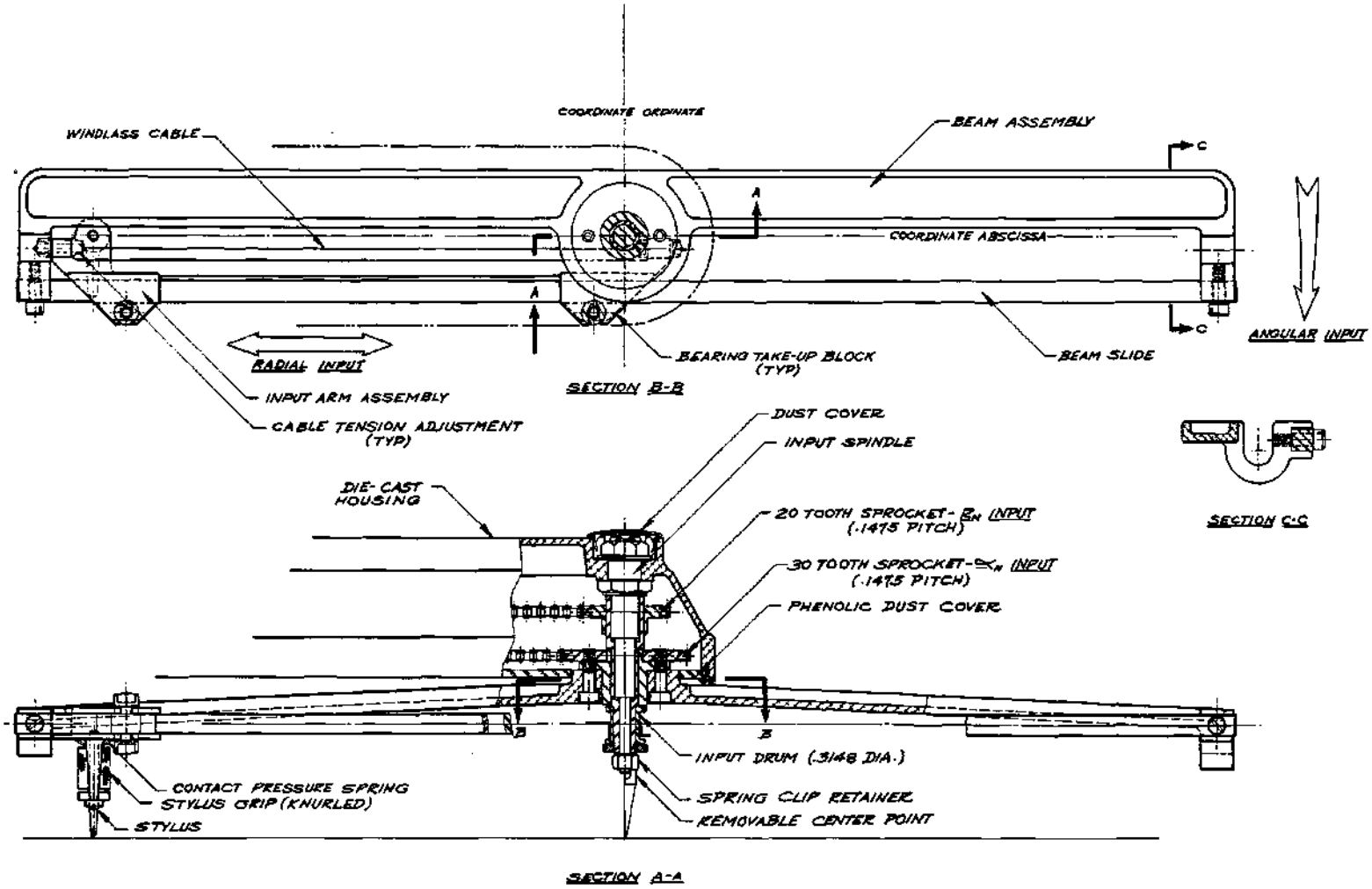


Figure 6. Stylus Input Assembly.

established directions of the ordinate and abscissa. The input arm assembly is free to traverse on the beam slide. The beam assembly itself is free to rotate about the spindle axis. The maximum travel either side of center of the stylus is six inches. Therefore, the assembly can accommodate a circular path from zero to 12 inches in diameter by removing the detachable center point once orientation is established. Notice that the input arm is a windlass arrangement that causes the 20 tooth sprocket to rotate as the arm traverses. This action, coupled with the rotation of the beam assembly exactly duplicates the trajectory as would a suitable polar vector, the tip of which is the stylus rotating clockwise around the origin. The 20 tooth sprocket inputs the (r_n) variable and the 30 tooth sprocket, affixed to the beam assembly, inputs (α_n) directly. The input arm cause the .3148 diameter input drum to rotate one revolution for each inch of radial trajectory change.

Now that (r_n) and (α_n) are in effect continuously "picked up" from the phase portrait through the stylus input assembly and both transformed into rotating variables at the spindle, it would appear that the inputs are proportioned and ready for the actual computing mechanisms. This however, is not entirely true. A closer look at Figure 6 reveals that as the beam assembly rotates, it inputs also to the windlass drum by reason of the fact that the line of action of the windlass cable does not pass through the axis of the input spindle. Notice that even if the trajectory is a circle, its center at the origin, this automatically takes place. For a circle, (r_n) is

constant, but a complete revolution of the stylus will input exactly one revolution to the (r_n) sprocket besides inputting the actual ($\propto_n$) change. Obviously this differential action of the mechanism must be compensated for.

The fact that this problem arose led to an easy solution of another aspect of the design. Differential action, in this case, could most readily be accomplished through gearing. A phase portrait-to-mechanism reduction could also be affected in the same train. It was decided in the design that a mechanism reduction of six-to-one be established. That is to say, the maximum trajectory radius of six inches would now appear in the computing mechanisms as a maximum of one inch.

A planetary gear train was chosen to give the required reduction and differential rotation. Figure 7 shows schematically the approach to the problem. The 45 tooth sprocket inputs rotation to the carrier arm to which is affixed the 72-32 compound sun gear. The 20 tooth sprocket inputs (r_n) directly to the 24 tooth pinion. A 64 tooth driven gear is the output of the train. In planetary trains the reduction ratio or train factor, $\underline{e}$, is given by

$$e = \frac{n_l - n_a}{n_f - n_a} = \frac{\text{T (drivers)}}{\text{T (driven)}} \qquad (3\text{-}1)$$

n_l = speed of the last gear in the train

n_f = speed of the first gear in the train

n_a = speed of the carrier arm

For computations, assume that $n_1 = o$, that is, the rotation of the 64 tooth gear is zero. This would be the condition if the trajectory were a circle and (r_n) was therefore a constant value. For a complete revolution of the beam assembly, n_f must revolve once as seen from previous discussion. The planet arm is connected via its sprocket by means of a suitable drive from the beam assembly such that its rotation is proportional to (α_n). Reduction is accomplished through the train according to Equation (3-1).

$$e = \frac{\pi r\,(\text{drivers})}{\pi r\,(\text{driven})} = \frac{24}{72} \times \frac{32}{64} = \frac{1}{6} \qquad (3\text{-}2)$$

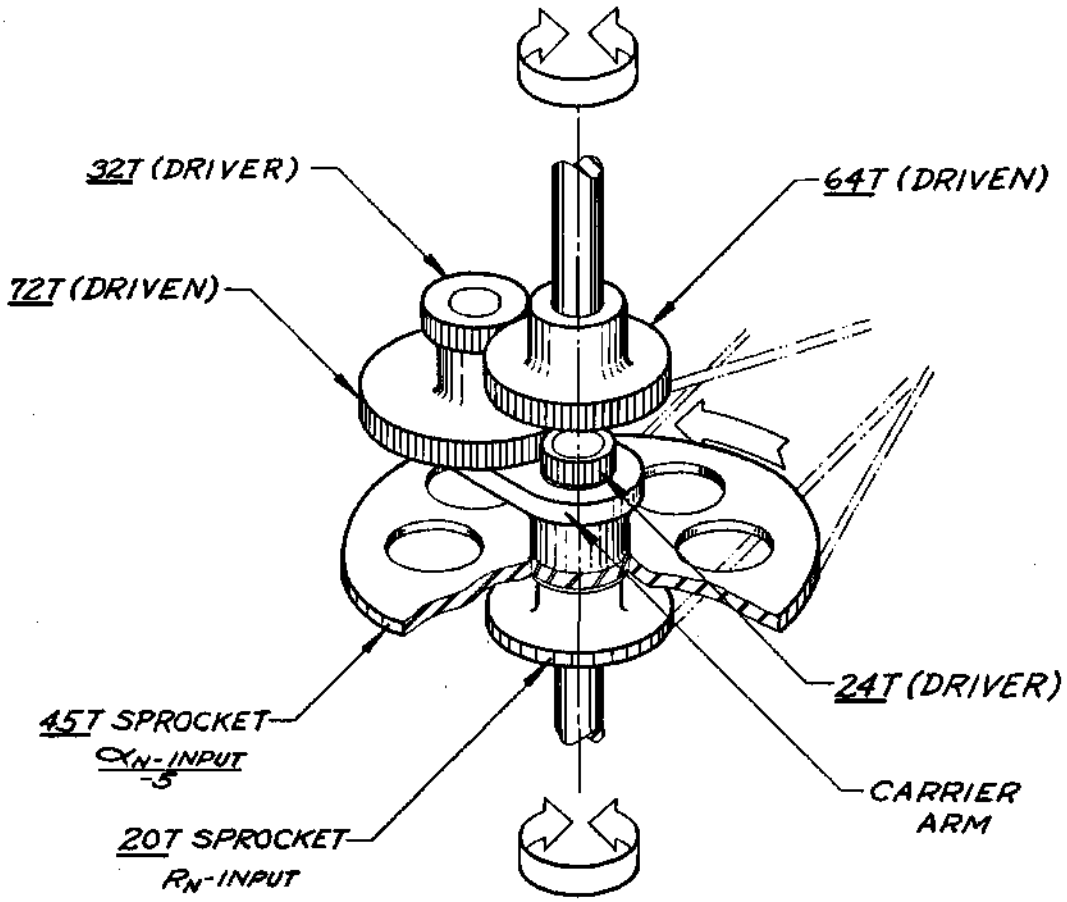
Now, solving for the speed of the arm, n_a,

$$\frac{1}{6} = \frac{o - n_a}{1 - n_a}$$

$$n_a = -\frac{1}{5}$$

Because of the nature of tracing a trajectory, the beam will always have clockwise rotation. Thus, the carrier arm must rotate at one-fifth beam speed and in the opposite or counterclockwise direction.

This assembly can be checked to see if it would actually be mathematically correct under all conditions that would arise in the general phase portrait. Suppose there is no (α_n) input. This would correspond to a radial line from the origin of the trajectory with α_n a constant value. Then the following results:



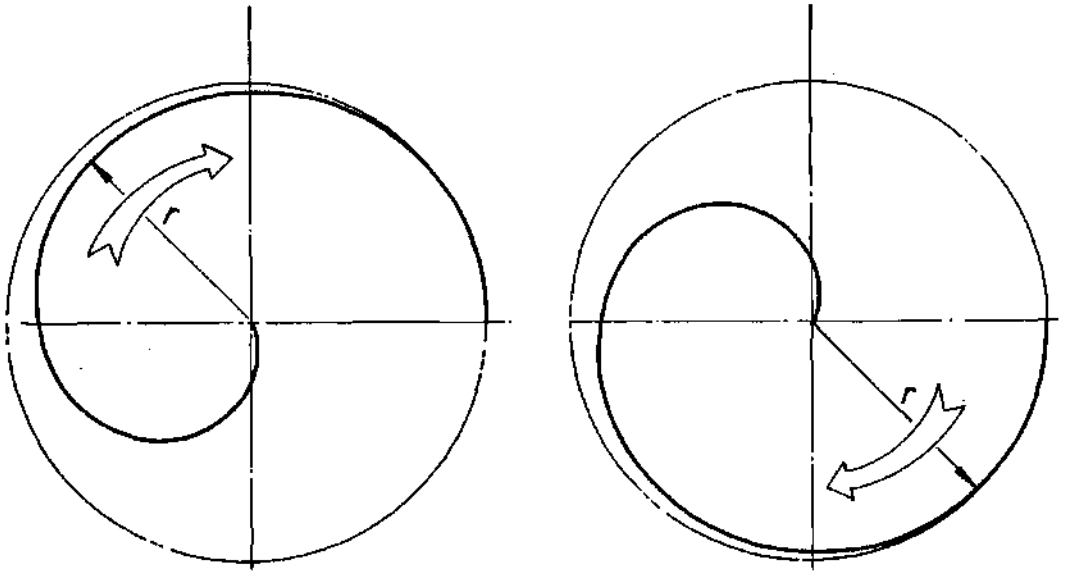
CRS

Figure 7. Schematic of Differential Input Epicyclic Train.

$$n_a = 0$$

$$e = \frac{n_1 - n_a}{n_f - n_a} = \frac{n_1}{n_f} = \frac{1}{6}$$

This obviously is the desired operation for this tracing. However, the circular and straight line trajectories are special cases. Assume now two general cases with some simplification to aid in computations. Figure 8 shows the two situations. In Case 1, the trajectory moves from the origin to the maximum radius of six inches in one revolution of the beam. From Figure 6 it is noted that the increasing magnitude of the vector would cause a rotation of the windlass drum of six clockwise revolutions in this case. Too, the differential input of the beam inputs one clockwise rotation. Therefore, if clockwise is assumed positive, and counterclockwise negative, the 20 tooth sprocket of the gear train sees an (r_n) input of (+7) revolutions. Suppose now in Case 2 the stylus traces a trajectory from its maximum of six inches to zero in one revolution of the beam. Again the beam rotation inputs one clockwise revolution. The (r_n) windlass however, has input six counterclockwise revolutions for a total of (-5) revolutions at the train input. As will be seen later, in the design of the function multipliers, it is necessary to have a clockwise rotation of n_1 in Case 1 and a counterclockwise rotation in Case 2. The following is a proof of correct operation of the differential under the two case conditions:



Case 1

Case 2

Figure 8. Two Case Solution of Input Rotation.

CRS

Case 1	Case 2
$n_1 = +1$ (assumed)	$n_1 = -1$ (assumed)
$n_f = +7$	$n_f = -5$
$n_a = -1/5$	$n_a = -1/5$

$$e = \frac{n_1 - n_a}{n_f - n_a}$$

$\frac{1}{6} = \frac{n_1 - (-1/5)}{7 - (-1/5)}$	$\frac{1}{6} = \frac{n_1 - (-1/5)}{-5 - (-1/5)}$
$\frac{6}{5} = n_1 + \frac{1}{5}$	$-\frac{4}{5} = n_1 + \frac{1}{5}$
$n_1 = +1$	$n_1 = -1$

In both cases, the rotation of the output of the train to the multipliers is that required and therefore the train functions as designed in all cases.

Figure 9 shows the finalized Differential Input Assembly. The transfer shaft accepts the (r_n) variable from the 20 tooth sprocket on the stylus input spindle. This particular arrangement is necessary for sprocket assembly in Figure 6. Too, it is a convenient tension adjustment for the instrument chain drives. The ($\propto_n$) variable is input directly to the 30 tooth sprocket on the sine-cosine cam plate shaft. Two 9 tooth idler sprockets run from the ($\propto_n$) chain. This arrangement drives the 45 tooth sprocket of the carrier arm. Observe that the rotation of the arm is one-fifth that of the cam plate sprocket and is in the opposite direction. The final differential output is available at the multiplier drum input shaft. The two discrete

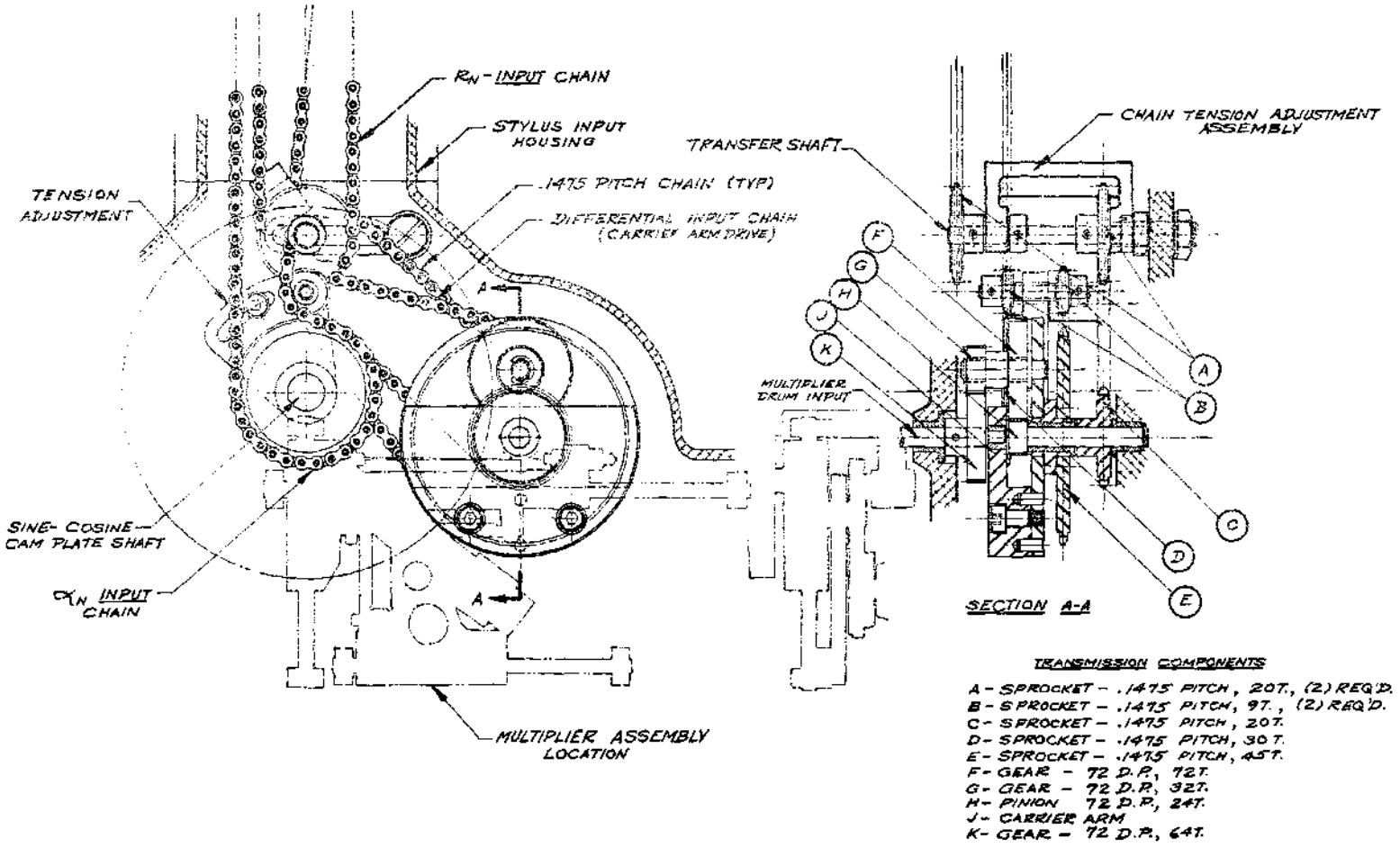


Figure 9. Differential Input Assembly.

variables are now in the form of (α_n), at the cam plate shaft, and a one-sixth reduced value of (r_n) at the multiplier drum shaft.

Sine-Cosine Function Multipliers

Once (α_n) and (r_n) are extracted from the phase trajectory, the approximating segments of the trajectory can be formed. With reference to Chapter II, iterative values of $(r_n) \sin\alpha_n$ and $(r_n) \cos\alpha_n$ must be computed. This operation involves two multiplications and the formation of the sine and cosine functions. Figure 10 shows analytically the functions of the multiplier (2). By similar traingles the following is true:

$$\frac{y}{c} = \frac{z}{x}$$

$$z = \frac{xy}{c} \tag{3-3}$$

Now if $y = (r_n)$, $x = \sin\alpha_n$ and c assumed equal to unity, then

$$z = (r_n) \sin\alpha_n$$

In like manner, for $x = \cos\alpha_n$,

$$z = (r_n) \cos\alpha_n$$

Figure 11 shows the final multiplier for the cosine function. The basic multiplier assembly is shown in phantom in Figure 10 to illustrate

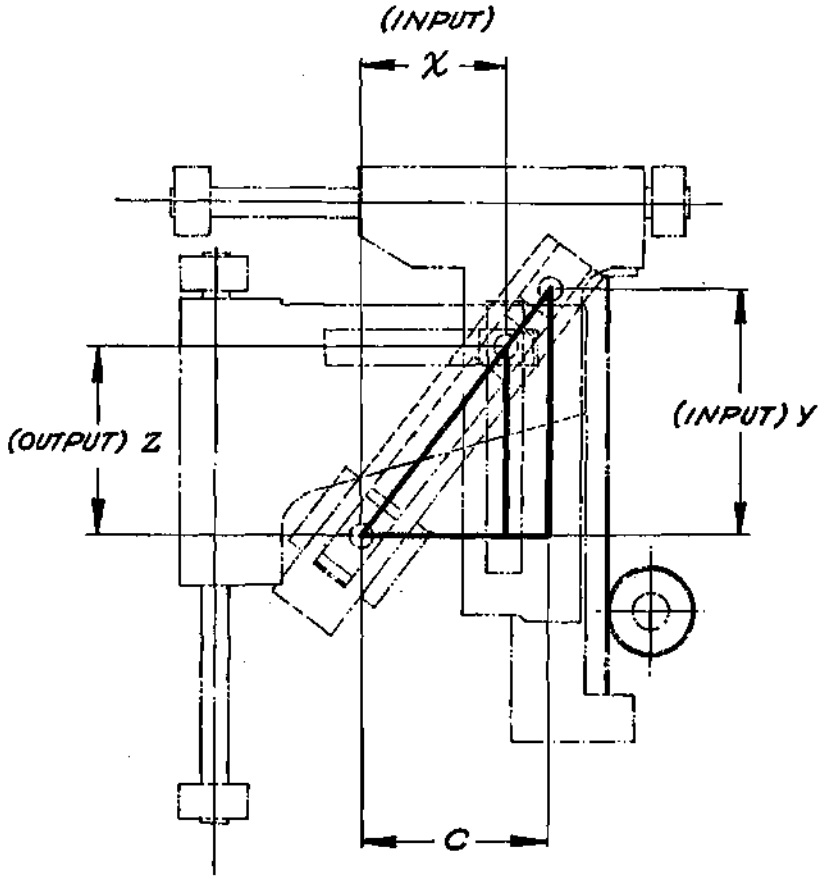


Figure 10. Multiplier Functional Configuration.

the mathematical feature of the mechanism. The multiplier in Figure 11 produces ($\sum \Delta x$) values. As noted on the figure, the sine function multiplier is similar but of opposite hand. In the final assembly, these two multipliers are stacked, the (b_n) or sine function multiplier over the ($\sum \Delta x$) mechanism in a sandwich fashion. The special two-faced sine-cosine input cam operates between these two assemblies as shown in phantom on Figure 11. The cam generates absolute values of sine and cosine since time is always positive. In fact all the variables are dealt with as positive quantities, notwithstanding their position on the phase portrait. Such a cam operates exactly like a scotch-yoke mechanism except that the variable, or slide, never passes through zero and therefore at no time is it negative. The multiplier drum receives direct input from the output of the differential input assembly. Both multipliers are designed so that one revolution of this drum is affected by a six inch maximum radial travel of the input stylus. The cosine multiplier drum is .4244 inch in diameter. This gives one inch of travel to the radial input slide at the connector block. The sine multiplier requires three-fourths inch travel with a .3183 inch diameter drum. Both cam plates give a trigonometric magnitude of three-fourths inch to the angular input slides. The plates are displaced 90° relative to each other on the cam shaft so that as the assembly is rotated by the (α_n) input, there is a correct trigonometric as well as angular correspondence between the two multipliers. This is to say that the sine cam plate moves its follower to its maximum magnitude when the corresponding position on the cosine plate is zero.

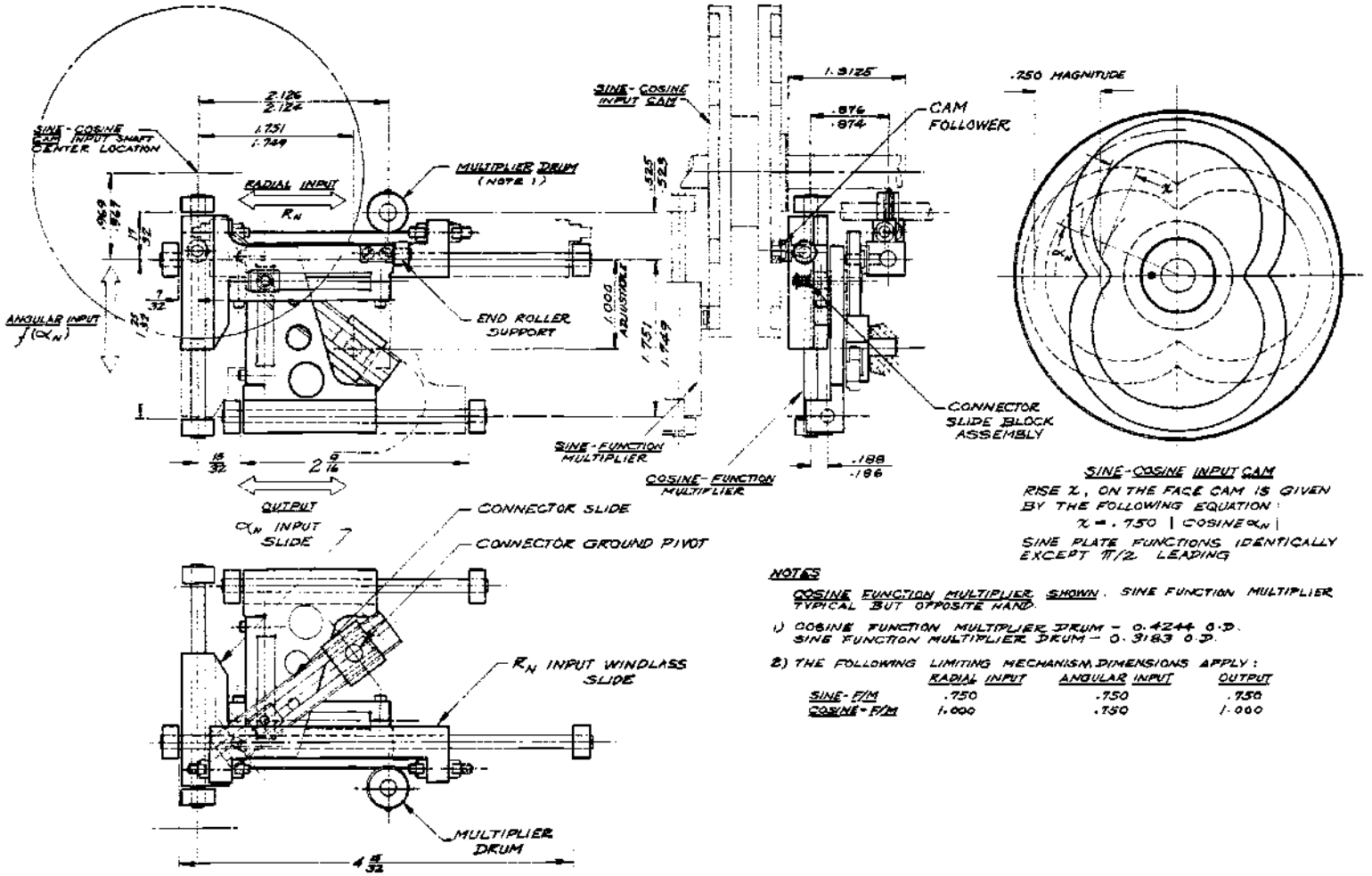


Figure 11. Function Multiplier Assembly.

As is obvious from the appearance of the device, with (α_n) and (r_n) as continuous inputs from the stylus tracing the trajectory, the component slides form a "collapsing" triangle mechanism. The central connector slide block assembly mechanically joins all three slides. Any variance in either or both of the input slides of a multiplier causes a proportional displacement of the output slide. Finally, the output of the cosine function multiplier becomes the input to the iteration timer, while the output of the sine function multiplier is directly connected to the iterative adder.

Iteration Timer and Iterative Adder

The iteration timer and iterative adder are the basic assemblies of the computer. These two mechanisms take the continuous outputs of both multipliers and give the remaining sequences of the computation a definite digital capability. Since the trajectory is approximated by straight line segments, it is necessary to use discrete values of the (b_n) and ($\sum \Delta x$) output variables to compute Δt values.

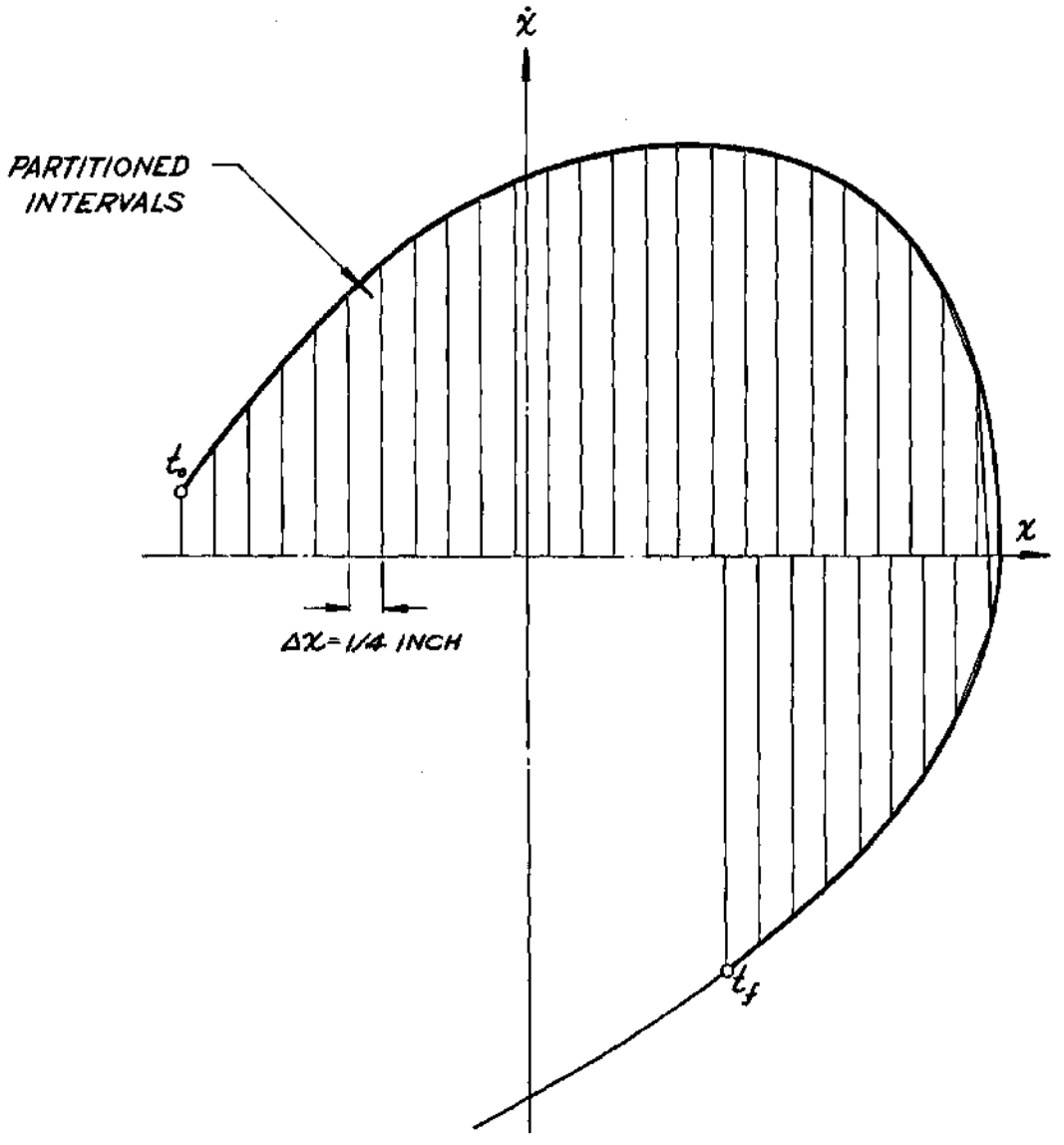
Before the actual mechanisms that perform these functions are reviewed, it is helpful to analyze the job they must accomplish. It was decided that all trajectory segments would project onto the abscissa at one-fourth inch intervals. This is actually saying that the computer will take the (r_n) $\cos\alpha_n$ output of the cosine function multiplier and, starting from the initial position of the trace, proceed to partition the area under the plot into one-fourth inch wide columns. This would continue for as many revolutions or parts of a revolution as would be desired to trace. Understand that no assumptions are made that half-

cycles of a trajectory are one-fourth inch multiples. If they are not, the computing mechanism does not notice. The computer simply sees the abscissa projection of the entire trajectory traced, no matter how many overlapping cycles are passed. The projected trace is one continuous length to be divided into one-fourth inch intervals. Figure 12 shows how part of a trajectory might be partitioned starting from an initial position at t_o and ending at t_f. The purpose of the iteration timer is to give mechanical signals to the iterative adder and elapsed time summer at each partition so that a value of Δt can be computed and stored during each interval.

The iterative adder, working in conjunction with signals from the timer, must form a value of average velocity at each interval. From Chapter II, the mechanizing equation is

$$b_n + b_{n-1} = \dot{x}_{avg-n}$$

Examination of this equation reveals that each previous value of (b_n) must be stored and used in computations during the Nth interval. It is only necessary to use this value of (b_{n-1}) during the Nth interval and therefore, the adder functions to change each (b_{n-1}) value to the (b_n) value after the addition is performed at each partition. In fact, a value of (b_{n-1}) for the Nth partition is the (b_n) of the $(n-1)$th interval. This is true on all except the initial interval where b_o is set into the computer by the initial position of the input tracing stylus.



CRS

Figure 12. Partitioning Process of Computer.

In review, the iteration timer will partition the trajectory automatically into predetermined segments. It will, by doing this, signal the iterative adder to store and add the corresponding (b_{n-1}) and (b_n) values. At the same time a Δt value is formed and is logged into the elapsed time summer.

Figures 13 and 15 comprise the basic assemblies of the timer. The mechanism in Figure 13 functions to partition the trajectory. The Sequence Cam Drive in turn powers the Iteration Timer shown in Figure 15.

The cosine function multiplier is partially shown in phantom in Figure 13. Notice that the output is directed to the input windlass of the drive train. Varying inputs to the multiplier cause the windlass to traverse within a maximum output range of one inch. The windlass drum is .3184 inch in diameter and therefore a full range output affects one revolution of the drum spindle A. The gear train from spindles A through E is of the step-up variety of eight-to-one ratio. The design of the train required consideration of the effects of inertia since rotational speeds are increased. Minimizing transmitted inertia is easily accomplished by use of a variety of standard equations or the results shown in Figure 14. The desirable number of meshes and the mesh ratios can be obtained from these two graphs (3). Remembering that backlash can always be a problem in instrument trains, the minimum number of meshes paractical should be chosen. As seen, mesh numbers higher than three offer little advantage in minimizing reflected inertia for the required ratio. From the nomogram of Figure 14 the

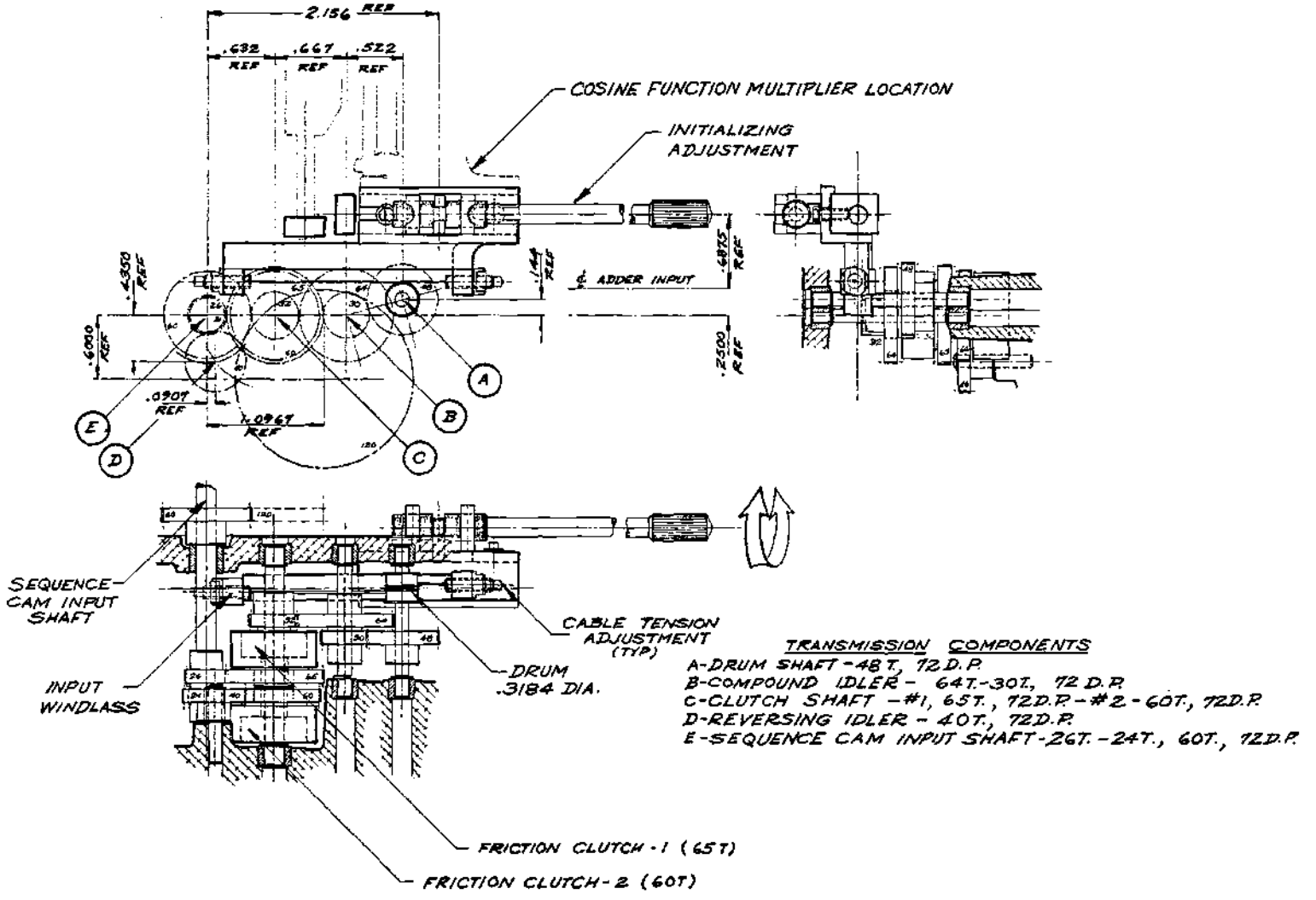


Figure 13. Sequence Cam Drive Assembly.

CRS

following ratios were chosen:

first mesh - 1.6 : 1

second mesh - 2.0 : 1

Of course the overall ratio divided by the two chosen mesh ratios gives the ratio of the third mesh, which is 2.5 : 1. Gearing is now chosen for physical mating and space requirement. The final design uses standard gearing conforming to American Gear Manufacturers Association specifications. The first mesh is made through shafts A and B by the 48 to 30 tooth pair. The B shaft is compound geared through a 64 - 32 tooth pair. The third mesh is compounded through two indexing friction clutches to 65 - 26 tooth and 60 - 24 tooth pairs. Notice that both 65 - 26 and 60 - 24 tooth pairs have the required third mesh ratio of 2.5 : 1. Since these gear sets are on the same centers, a 40 tooth idler is required to mate the 60 - 24 pair. The idler introduces a direction change in this path of the third mesh. It was realized in prior explanations that the partitioning feature of the analysis would recognize all $(r_n)\cos\alpha_n$ projections as positive quantities and segment them with no regard as to which quadrant the trajectory trace was in. Also it was noted that the input windlass would cause both clockwise and counterclockwise rotations of the drum shaft since in general the phase plot would pass through the $\alpha_n = \pi/2$ and $\alpha_n = 3\pi/2$ positions in any 2π rotation. It is necessary to design into the train the capability of a clockwise rotation of the sequence cam input shaft no

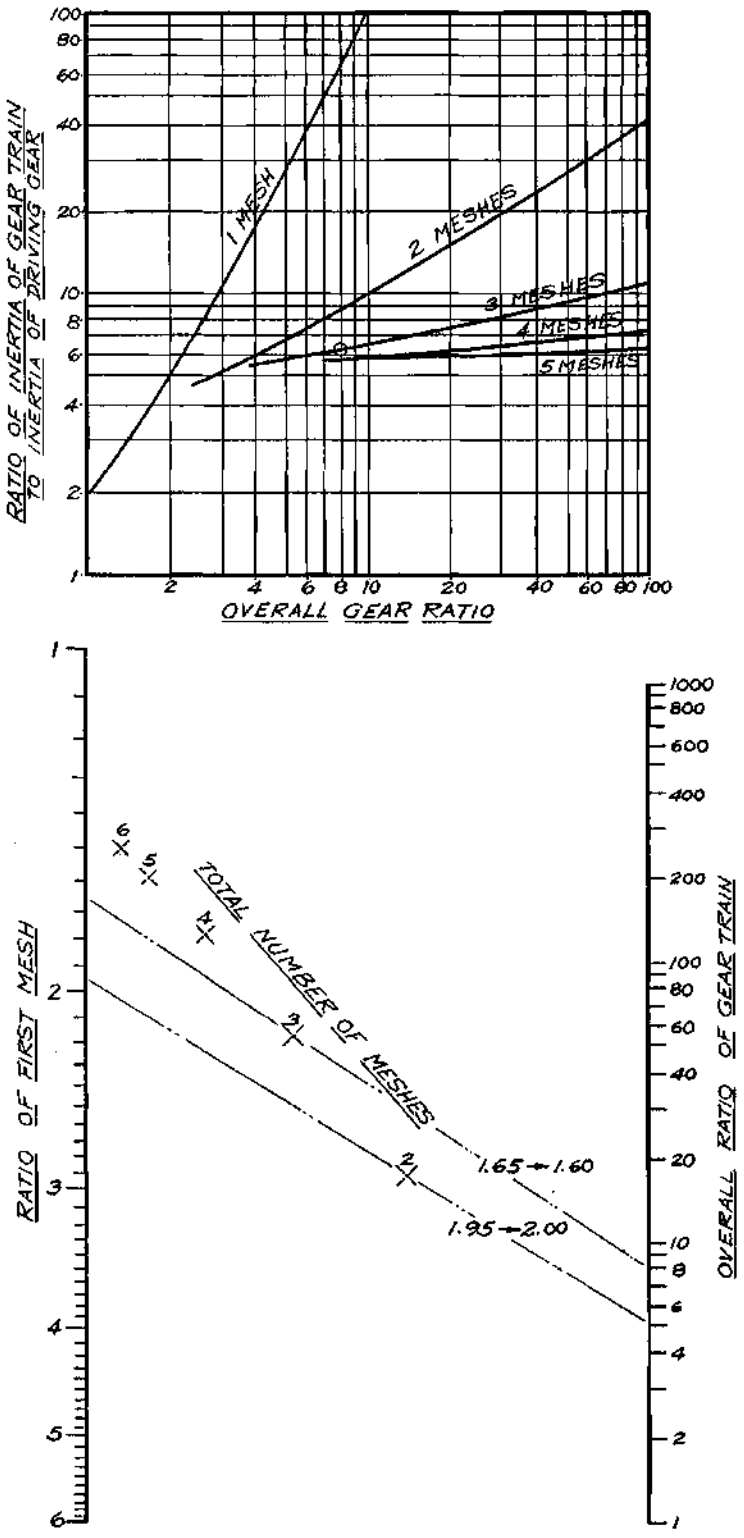


Figure 14. Reflected Inertia and Mesh Ratio Selection.

CRS

matter the rotational direction of shaft A. The two gear pairs and clutches on the third mesh of the train give the mechanism this feature of absolute value generation. Notice that as the sequence cam shaft rotates clockwise, the meshing drive can be traced back through the train to shaft A by two separate paths. First, by way of the 26 - 65 tooth mesh, the drum rotates counterclockwise; however, through the 24 - 60 tooth mesh, considering the 40 tooth idler, the drum shaft must now rotate clockwise. The two clutches allow selection of the proper mesh path by virtue of the direction of rotation of the drum. The clutches are commercially available assemblies that drive positively in one direction of rotation but are free in the opposite direction. Consider now that, at an instant, the windlass drum has clockwise rotation to complete the train through the 60 - 40 - 24 tooth mesh for clockwise rotation of the cam shaft. Clutch 1 cannot drive in the clockwise direction and therefore it freewheels when shaft A has clockwise rotation. Conversely, if the input drum rotates counterclockwise, shaft C also rotates counterclockwise necessitating clutch 1 to drive while clutch 2 is free. Now rotation is through the 65 - 26 tooth mesh and the sequence cam shaft again has clockwise rotation. It is important to note that since the ratio between the 26 - 65 and 24 - 60 meshes is the same, no relative rotational velocity exists between the clutch assemblies that would cause them to drive simultaneously.

The iteration timer and the iterative adder work in correspondence to produce the average values of velocity at each Δx interval. The iteration timer is an assembly of two cam actuated locks. As previously

illustrated, a storage feature is employed in the adder to accomplish the summation of consecutive values of (b_n) and (b_{n-1}). It is necessary in the adder to "lock in" each value of (b_{n-1}) during the increment period. This period is designed to be timed by the iteration timer operating through the cosine function multiplier and the sequence cam drive assembly. The step-up gearing of the cam drive increased the input drum rotation by a factor of eight. Figure 15 shows the sequence cam drive shaft geared to the lock plate assembly through a 60 - 120 tooth set. There are two cam plates, one for each lock, and each plate has six equally spaced lobes. As shown, the design of the cosine function multiplier provides for one complete revolution of the input drum of the sequence cam train for a maximum stylus travel of six inches. Therefore, if each lobe of the cam plate functions per an interval, then the product of the number of lobes and the multiplying factor of the gear train is 48. This number is divided by the reduction of the 60 - 120 tooth mesh from cam shaft to plate assembly which gives 24 intervals in six inches of trajectory radius. This follows from the one-fourth inch partitions in the design specifications. Notice that the cam plate gear set gives the mechanism a degree of flexibility. The 60 - 120 set could be replaced by two 90 tooth gears for a one-to-one ratio and one-eighth inch partitions. Obviously, a number of ratios can be provided for by a collection of change gears. The analyst then has a variety of partition sizes so that a mechanism-to-trajectory ratio can be changed if desired.

In Figure 15, the locks are designated #1 and #2, each activated

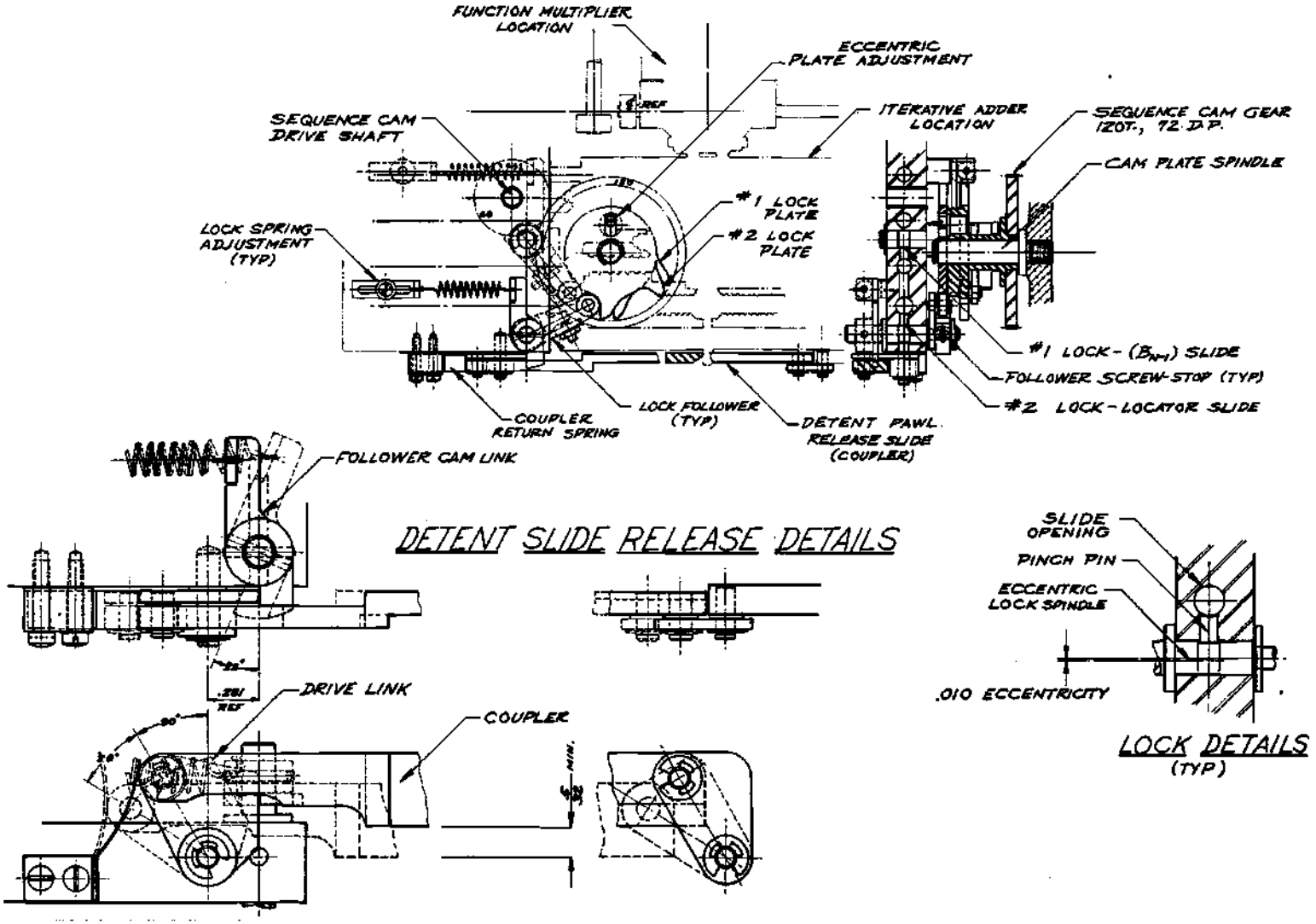


Figure 15. Iteration Timer Assembly.

by the corresponding cam plate. A discussion of the iterative adder in Figure 16 will illustrate the use of these locks in precisely controlling the summation. The sine function multiplier's output slide is connected through the adjustable input tongue assembly to the (b_n) slide of the adder. The summation

$$\frac{b_n + b_{n-1}}{2} = \dot{x}_{avg-n}$$

is exactly computed by the adder mechanism schematically shown with the assembly. In operation, imagine that the two windlass slides designated X and Y are racks with the windlass drum a pinion meshing with both. The axis of the pinion is affixed to a traversing output shaft, Z. Suppose that the Y rack, for example, is assumed fixed while the pinion moves a distance Z. Then the X rack must move a distance Z plus the displacement imparted by the rotation of the pinion which is of course Z also. Therefore, any displacement of the pinion slide is actually half that of the rack. Now if both (b_n) and (b_{n-1}) racks are displaced, they have relative motion between the ground link and themselves and each can be considered fixed with respect to the other. It follows that the pinion is displaced one-half the sum of the two input side displacements (4). More specifically,

$$Z = \frac{(X + Y)}{2} = \frac{(b_n + b_{n-1})}{2} = \dot{x}_{avg-n}$$

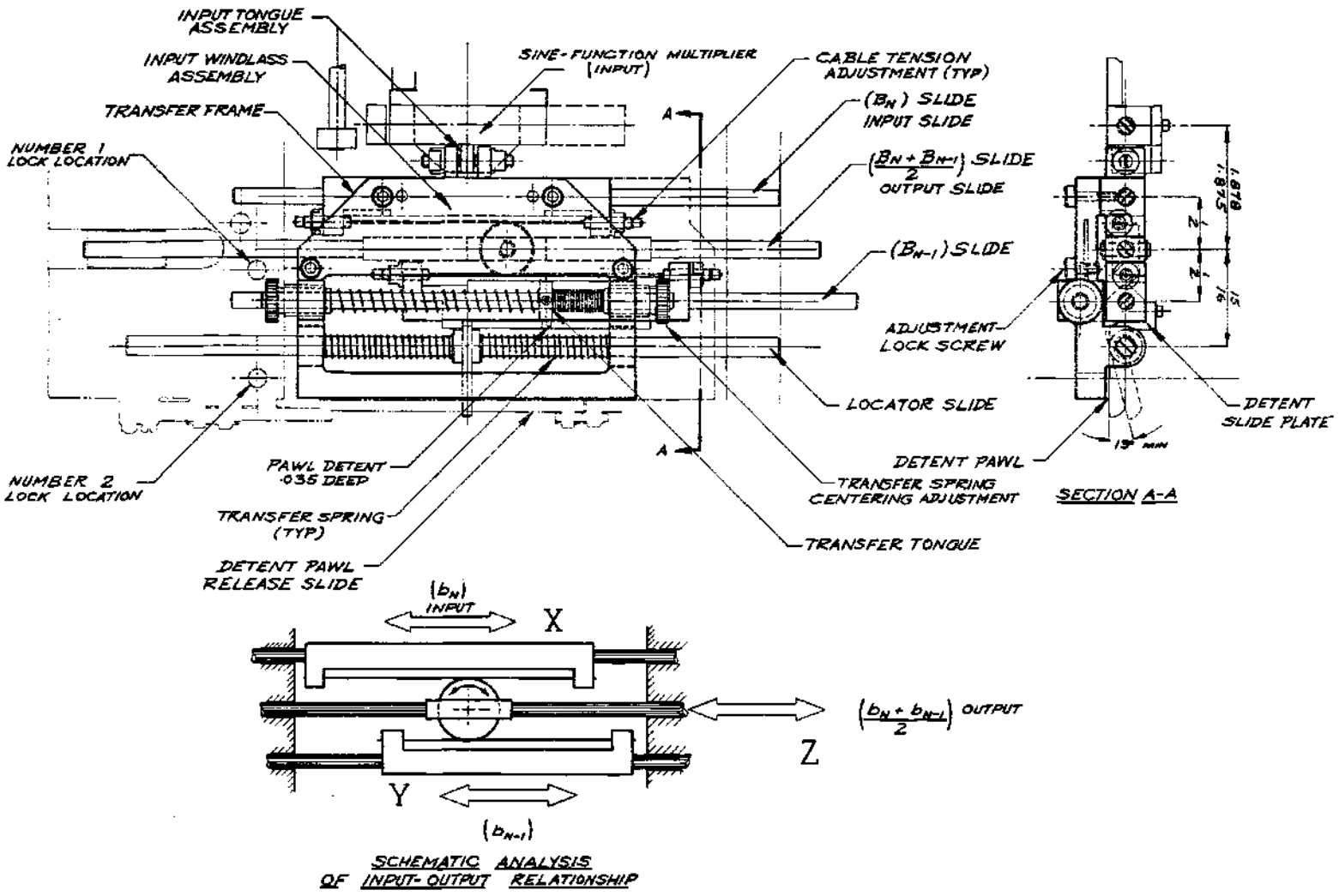


Figure 16. Iterative Adder Assembly.

which is the approximating sum desired. In this mechanism, windlass and drum assemblies are again used to eliminate backlash.

The input to the (b_n) slide is continuous since it is directly coupled to the sine function multiplier output. The (b_{n-1}) slide is mechanically connected to this output through the transfer frame and springs. The transfer springs are identical, so their natural tendency is to center the transfer tongue with the central axis of the frame. Notice in Figure 16 that the tongue is displaced to the right of the frame's center. Were it not for other features of the assembly, the (b_{n-1}) slide would travel with the (b_n) slide always giving the output a displacement of (b_n). Adjustable stops, denoted as transfer spring centering adjustments are used on the spring shaft to allow the tongue to be brought into exact floating alignment.

The (b_{n-1}) slide can be locked to the computer frame, or ground, by Lock #1. When this happens, the (b_n) slide is free to move since the transfer springs will compress or extend within the transfer frame. If the lock is suddenly released, the (b_{n-1}) slide will snap into the center of the frame to assume a coincident position with the (b_n) slide. A second spring loaded assembly, designated the locator slide, operates in a similar manner to the (b_{n-1}) slide. Like the (b_{n-1}) assembly, this slide has its ends extending into the ground frame and can be maintained in a given position by Lock #2. The locator slide carries a centrally located detent pawl that can rotate a fraction of a turn within the confines of the transfer frame. The springs on this slide have one end pinned into the transfer frame and the other into the

respective side of the pawl. These springs are loaded in torsion to keep the pawl face in contact with smooth surface of the detent slide plate. This detail has a shallow, precision female detent located in line with the transfer tongue that mates exactly with the nose of the pawl. In operation, if the pawl is riding on the slide surface and passes over the catch, it will drop into the detent and form a mechanical connection between the (b_{n-1}) and locator slides. Therefore, if the pawl is mated with the catch on the (b_{n-1}) slide, and the locator assembly is locked to ground by Lock #2, then it is also true that the (b_{n-1}) slide is grounded to the computer frame.

Each lock is an assembly of an eccentric cam which acts against a pinch pin to wedge the slide spindle to its bearing in the ground frame. This detail is shown in Figure 15. A small rotation of the eccentric causes a large binding force. Locks of this type are quite positive and are used in applications such as locking a micrometer spindle to its frame.

To clearly realize the operation of the iterative adder, it is best to actually proceed through a summation sequence of (b_n) and (b_{n-1}) values from an initial position on the trajectory. For now, the discussion will not dwell on the precise timing of the cam plates, but instead, the locking sequence itself, as part of the summing action, will be discussed. At an initial position of the stylus, the locator slide is locked by #2. As the stylus traces the general trajectory, #1 unlocks allowing the (b_{n-1}) slide to be positioned by the pawl at the initial (b_n) orientation. The (b_{n-1}) slide is in position (b_o).

Now the mechanism functions to lock #1. This action takes place with further movement of the stylus along the trajectory. Lock #2 now snaps open and the stored energy in the locator transfer springs pushes the slide into correspondence with the (b_n) slide. The locator and (b_n) assemblies now have the same instantaneous value of the trajectory projection, and this condition will exist as long as Lock #2 remains open. Now with a slight further movement of the stylus, #2 locks and the entire sequence is ready to be repeated. This cycle of mechanism events takes place in exactly one interval of Δx partitioning. Reviewing, at the start of the interval, #2 is locked, #1 unlocks an instant later, and the (b_{n-1}) slide assumes the (b_o) position of the locator slide. Lock #1 then locks and #2 unlocks allowing the locator to assume the position of the (b_n) slide. Precisely at the end of the Δx interval, #2 locks and the locator slide has secured the (b_1) value. The sequence will repeat itself as many times as there are Δx intervals traversed on the trajectory. The output of the adder has as its instantaneous value the average velocity for the partitioned interval. As #2 locks, this value is received by the divider to produce a Δt for the cycle. Although not mentioned in the prior analysis, the detent pawl is released from the catch to allow the (b_{n-1}) slide to assume position during each cycle. Detailed discussion of this mechanism will follow.

The timing for the locking sequence is very important. The design calls for a careful selection of the transfer springs and other dynamic details of the assembly since the sequencing will usually be fast.

The actual speed, of course, is entirely dependent on the rapidity of tracing the trajectory. Figure 17 shows exactly the sequence of mechanism events that occur during an interval of summation.

The timing of the locking and unlocking of both slide assemblies is controlled by the lock plates in Figure 15. The relative positioning of the two plates with respect to each other will determine the interval of locking of each slide within the partition. This relationship is changed by means of the eccentric plate adjustment. With counter-clockwise rotation of the cam plates, a lock opens as the follower rides up the knee of the lobe and snaps closed as it sharply falls off the rise. The #1 cam plate operates only the (b_{n-1}) slide lock. However, the #2 cam plate both locks the locator slide and actuates the detent pawl release. The #2 lock follower has attached an integral camming lug that operates the release as shown in the details of Figure 15. A 23^{o} rotation of the follower is sufficient to cause the drive link to rotate and carry the coupler bar down with parallel motion. The leaf spring helps return the linkage to its original position. Reference to Figure 16 shows that at any position within the range of the adder, the detent pawl is pushed down by the release coupler, simultaneously releasing the locator slide from the locked (b_{n-1}) position. The slide surface of the coupler is polished to offer little resistance to the detent pawl as the locator assembly positions itself coincident with the (b_{n}) member.

Divider and Elapsed Time Summer

To this point, the computing mechanisms have resolved the

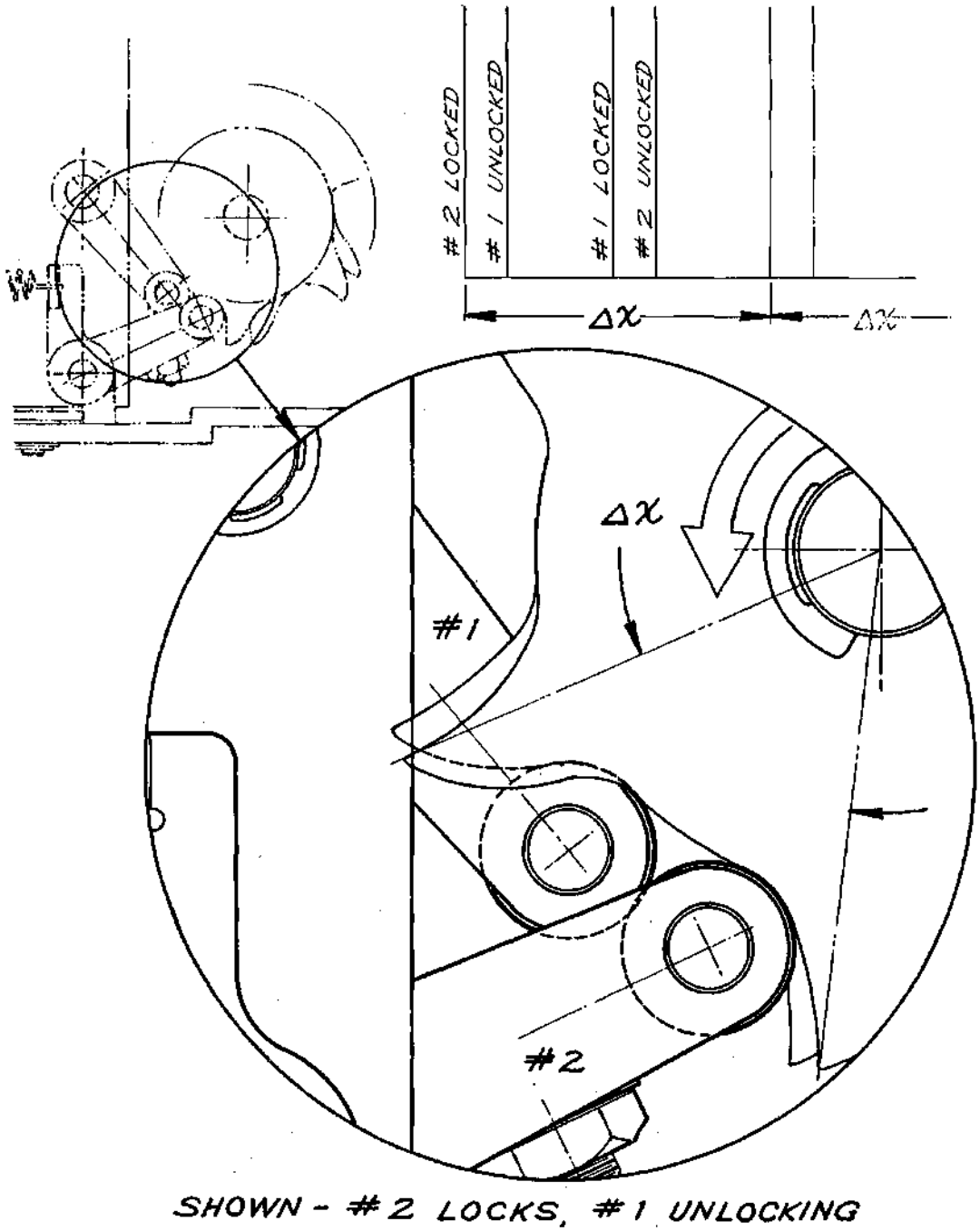


Figure 17. Mechanical Sequence of Addition Functions.

CRS

trajectory into polar components at partition boundaries and used these projections to form incremental values of average velocity. The fourth and final assembly of the computing train is the Divider and Elapsed Time Summer. The divider itself is merely a special formed cam. The time summer accepts each incremental value of Δt as output from the divider cam and stores these as a summed sequence. Unlike the maximum size restriction placed on the trajectory, the elapsed time is not limited. Once the mechanism capacity is reached, it can be reset to zero, the previous value of elapsed time recorded, and the tracing continued with all subsequent values added to the previous total. The summer is driven from the sequence cam drive and works in exact correspondence with the iterative functions of the adder.

Figure 18 shows the divider and summer mechanisms in detail. A drum and windlass arrangement is again employed to convert a maximum $\dot{x}_{avg-n}$ output of the adder of three-fourth inch to a half revolution rotation of the drum. Remembering that the partitions of the trajectory are all one-fourth inch intervals eliminates Δx as a variable. The rise or form of the cam is entirely a function of the average velocity. Obviously there is a relationship between a linear displacement of the adder output and the angular position of the drum or divider cam plate. Since a cam rotation of π radians matches the maximum output of the adder, any number of discrete angular and linear positions between zero and maximum output can be used as base points for actually constructing the cam rise. If however, the size of the Δx partition changes, as was mentioned in the discussion of the sequence cam drive,

a new cam profile would result due to the change in the constant factor or partitioning.

From Figure 18, note that the purpose of the input cam is to move the input follower slide assembly. Each of its lobes represents an input of one value of Δt during an iteration cycle. It is driven by direct attachment to the sequence cam drive through a shaft extension. The input cam rotates at twice the speed of the adder lock plates, thus the three lobes instead of six. The dividing cam, physically located under the input cam, rotates on the sequence cam drive shaft. A side view of the assembly clearly shows that the follower rollers contact simultaneously both divider and input cams. Since an output of the adder yields an angular position of the divider cam proportional to Δt, the linear travel of the follower must likewise be the value Δt as called for by the specific cam rise. It has been shown that $\dot{x}_{avg-n}$ may never be zero as the divider sees it, although a minimum value has been decided upon. Incremental values of time will be stored as linear displacements of the follower in direct relationship to the rise of the divider cam. While the divider cam positions the follower, the input cam moves it the extent of its travel, or back to the Δt_{min} position each time a lobe rotates past the roller contacts. Thus at each partition interval, a Δt value is received by the follower and input to the elapsed time summer.

The summer is simply a long range dial indicator where fractional or decimal parts of the inch are converted to time increments. The spindle extension passes through two single acting locks. The ground

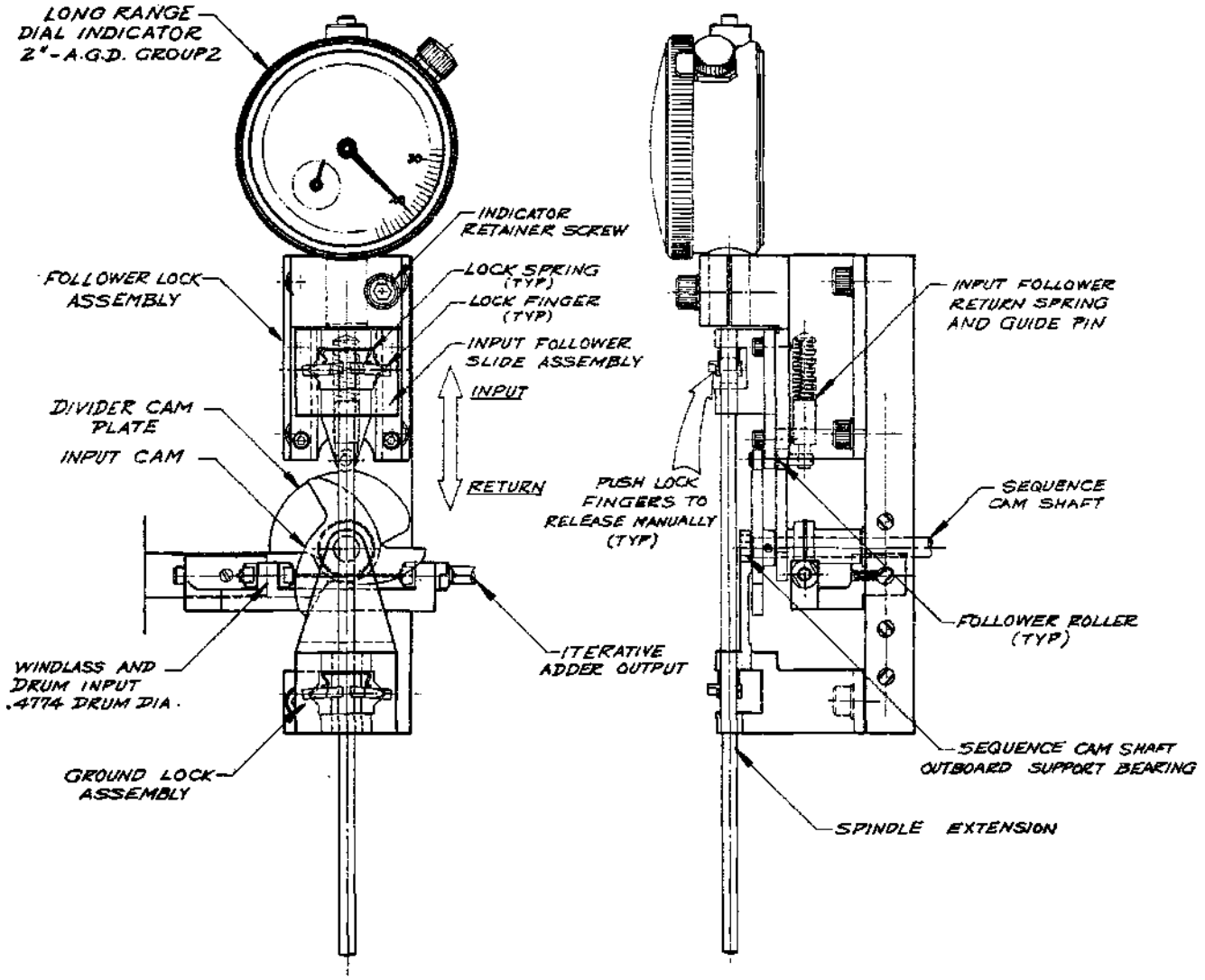


Figure 18. Divider and Elapsed Time Summer Assembly.

lock assembly maintains the indicator spindle in a set position, locked to the computer frame. The follower lock assembly allows the follower to move freely over the spindle extension in the "return" direction, but grips the spindle firmly when the input cam forces the follower in the "input" direction. In operation the spindle of the indicator is moved at Δt amount by the input follower at each partition interval or cycle. Once the input is complete, the follower return spring forces the follower to contact the dividing cam and the sequence is ready for the next Δt increment. The displacement of the indicator spindle, however, is maintained by the ground lock. As Δt increments are repeatedly input by tracing the trajectory, the total displacement of the indicator continues to depict the amount of elapsed time. A manual release on both locks allows for resetting the indicator to zero for extended time intervals or at the start of each new trace.

From Figure 19, an increment of time Δt_n is stored on the indicator only as the next value of $\dot{x}_{avg-n}$ is being computed. This illustration clearly shows the dividing cam and input cam as they function to store values of incremented time. The average velocity of each interval can be approximated only when the endpoint of each partition is obtained by the mechanisms. This is done when the stylus has traced through the particular Δx interval. Logically then, the storage of Δt for this interval must take place during the computation of the next value of $\dot{x}_{avg-n}$. In fact, each value of Δt stored is that of the preceding interval. It follows that if the trace is ended exactly on a partition endpoint, the Δt for this interval is not stored, and the total elapsed time of the said trace is the indicated value plus the

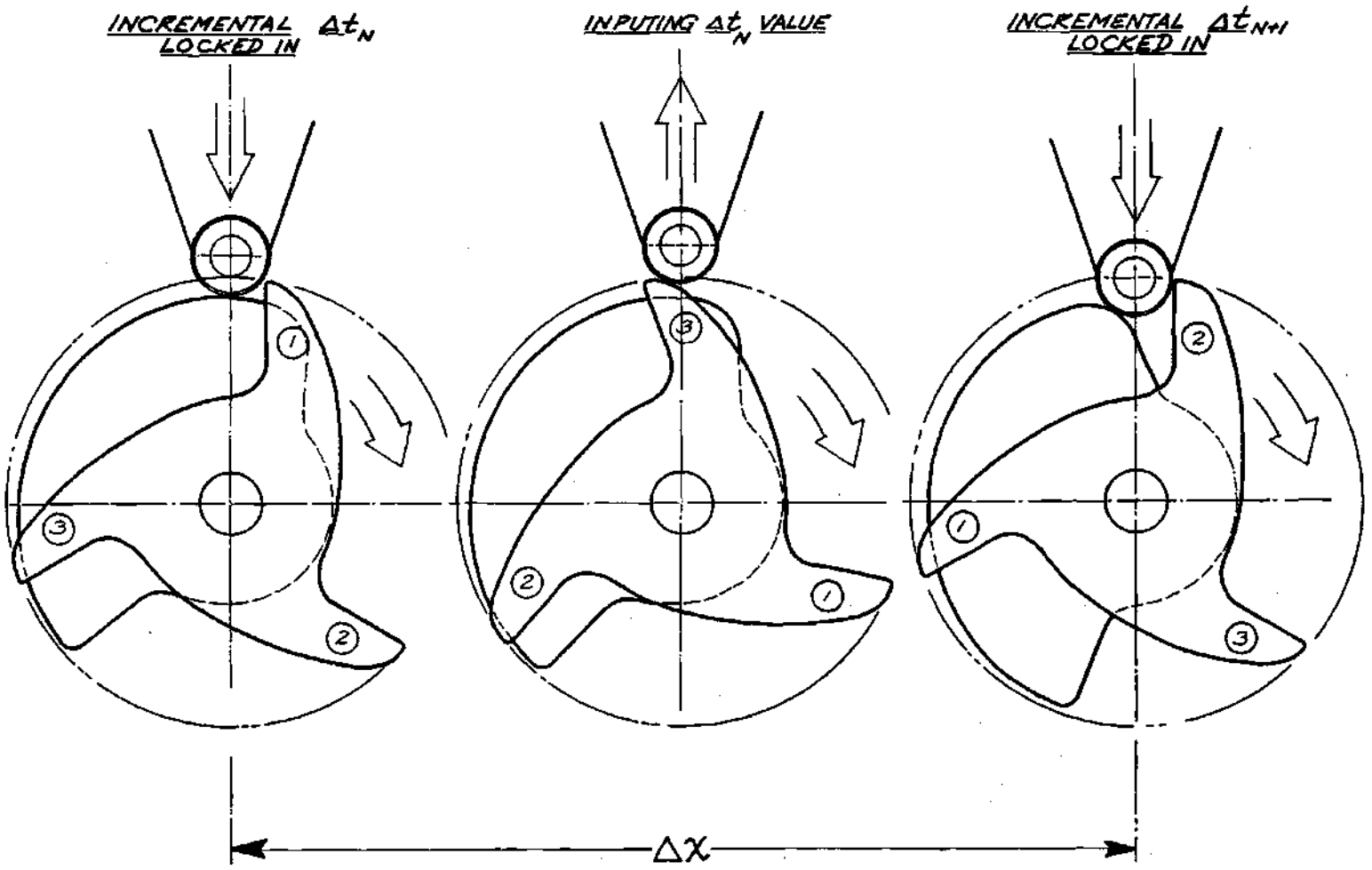


Figure 19. Mechanical Sequence of Time Summing Functions.

final unknown Δt increment. In general, the trace will not terminate exactly on an interval boundary. The majority of the traces will end somewhere within a partitioned area. The problem, however, is still the same. Part of the total time, namely that of the last interval, is left unindicated on the summer. The solution, though, is a simple one. The input cam of the summer can be used as a pointer to indicate partition boundaries. Observe from Figure 19 that Δx endpoints are always coincident with the sharp fall of each cam lobe. The user of the computer simply observes the cam and watches one or two Δx intervals elapse past the desired finish of the trace. At a partition endpoint, he records the value of the elapsed time on the indicator and the values of x and $\dot{x}$ one Δx interval back on the trace. He does this for one or more intervals before and past the trace finish. A quick plot is made of $\underline{t}$ vs $\underline{x}$ or $\underline{t}$ vs $\underline{\dot{x}}$ of these values around the finish point. Then the required end of the trace can be taken from the plot with the approximated value of t corresponding to this point. Any trajectory should lend itself to this procedure. In general, plots of limit cycles or conditions of unboundedness can always be traced past an endpoint to secure the final value of Δt.

Structural Consideration

Very definite calculations are important on the entire mechanism of the computer. Because it is mechanical and self contained, it possesses flexibility and utility that no present electrical devices have. Yet because it is composed of mechanical assemblies having inertial, frictional, and other inherent complexities, special attention

must be paid to materials, bearings, loadings and all the basic considerations of any mechanical design. There are components, especially in the iterative assemblies such as the adder and elapsed time summer, that operate with relatively high linear and rotational velocities. Where possible, such mechanisms are of light material or constructed smaller and of stronger metals. It should be apparent that many of the assemblies must be made with accuracy, while others do not necessarily require special precision. A commercially available device would no doubt employ die castings, stampings, sintered metal parts, and plastic injection molded pieces when possible. All the gearing, bearings, and other hardware are commercially available items at this time. It is reasonable to assume that the computer, as a production item of carefully tooled manufacturing, would be economically feasible for the purpose for which it is intended.

CHAPTER IV

OPERATING THE COMPUTER

The operation of the computer requires only the alignment of the plot with the axis of the input assembly, initializing the variables, and finally guiding the stylus along the line of the trajectory.

Referring to Figure 20, the computer is shown fastened to a suitable flat surface. The origin of the plot is placed directly under the input spindle and centered with the aid of the removable center finder. At the same time the abscissa is oriented coincident with a line connecting the center finder and a stop face at the base of the input assembly housing. The line is actually an axis of mechanical alignment perpendicular to the multipliers that automatically positions the mechanisms relative to the phase portrait. This orientation established the fixed relationship between the coordinate axis and the sine-cosine cam. This line may be drawn on the board if the installation is permanent, or a straight edge may be used to aid in the process. Having secured the plot to the board with tacks or tape, the computer is ready for initializing.

The stylus is positioned at $x = x_o$ and $\dot{x} = \dot{x}_o$. The velocity variable is automatically initialized. It is necessary to start the Δx intervals at an endpoint and therefore, the initializing adjustment (See Figure 13) is used to position the lock plates and the time

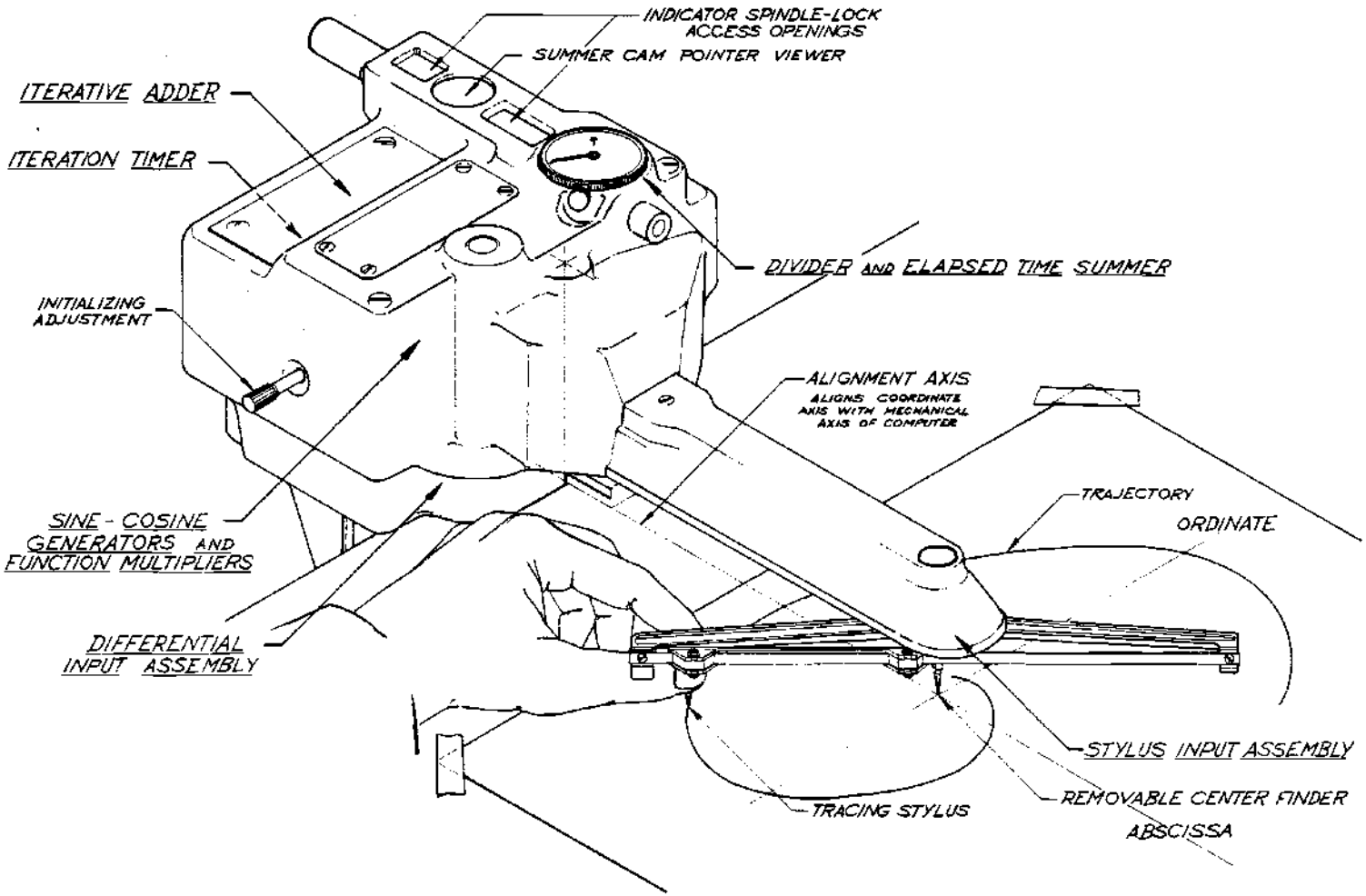


Figure 20. Operative Prototype Computer.

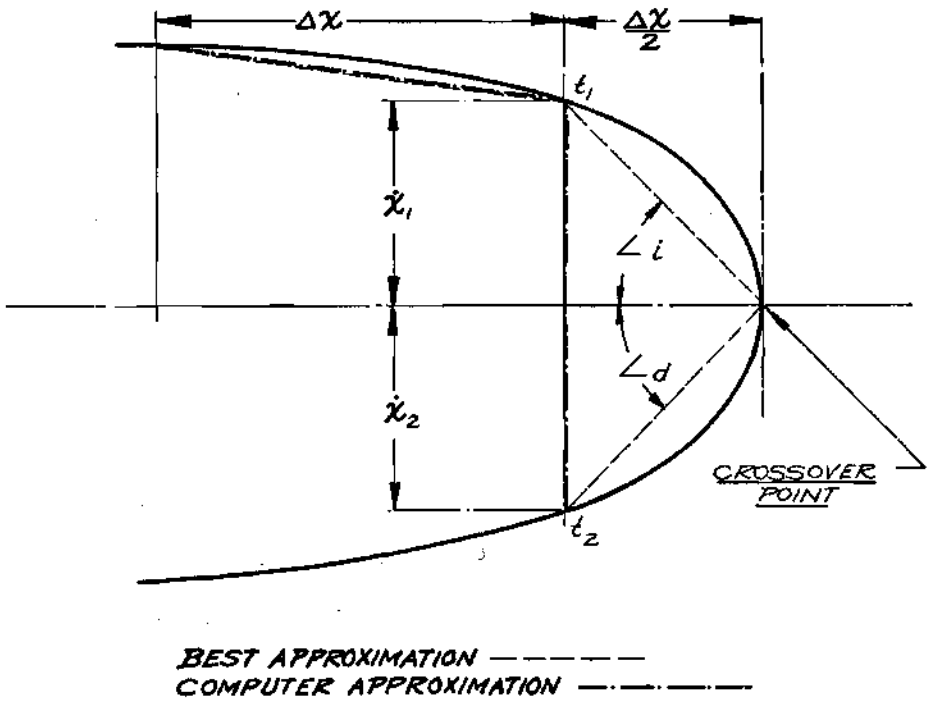
summer input cam to their boundary positions. This screw adjustment repositions the windlass within its range. Turn the adjustment in the appropriate direction until the summer input cam is seen to be at an interval boundary as shown in Figure 19. The displacement input is now assured of starting on a Δx endpoint. The Δt summer is now readied by simultaneously depressing the follower and ground lock releases as indicated in Figure 18. The indicator is now zeroed and the computer is ready to record the trace. The stylus is moved along the trajectory as far as desired, and the elapsed time is read on the dial of the indicator. If the portrait, or parts of it, have been drawn incorporating a scale factor, then the elapsed time is divided or multiplied, whichever is appropriate for the solution. Too, $\underline{x}$ vs $\underline{t}$ or $\underline{\dot{x}}$ vs $\underline{t}$ comparisons are readily obtained by watching the summer input cam and observing values at interval endpoints, then plotting them directly for response characteristics.

CHAPTER V

ERROR ANALYSIS

To be valuable to the analyst, the use of the computer certainly should be as accurate a procedure as any of the customary graphical techniques. If not, then in most cases, the advantages of speed and ease of operation are useless. Certainly there will be some error due to mechanism conditions such as backlash, clearances, and necessary manufacturing tolerances. However, these all can be controlled and minimized. It is necessary to assume that these inaccuracies will cause only a small percentage error that can be disregarded. Therefore, hypothesizing that the computer functions properly as designed, analysis can be made of any error which must result from either or both of two causes. The first problem would logically be an inadequacy of the computer to handle the general trajectory in every respect. This could point to flaws in the design and a possible failure to meet the project objective. The second cause for error would stem from the operator's inability to follow the trajectory accurately. If the computer functions as designed and the plot is traced exactly, the assumption was that the approximated answer would be accurate. For sake of discussion, the two possible causes of error must be dealt with separately since this is the only way to realize the effect of each.

The computer was designed to accommodate any general trajectory configuration. As discussed, a scale factor made on the plot can be used to alter slopes of the trajectory at the crossover points on the abscissa if these appear outside the range of the computer. However, scale factor changes would not be necessary with the majority of the portraits. Actually, the only way in which the computer receives information from the plot that is different from the basic graphical scheme is the use of a constant value of Δx. Graphically, the increments of x can be equal, but the last division is always conveniently made at the crossover of the trajectory at each half cycle. The computer does not handle the partitioning in this manner, but rather as shown in Figure 12. Therefore, as Figure 21 shows, the computer will not take the best approximation of the curve if the endpoint of the interval and the crossover point are not coincident. The computing mechanism was designed to accommodate a maximum angle of departure or incidence at crossover of 45°. Notice that as this angle decreases to zero, the Δt for the best approximation becomes unbounded. Too, the difference in the approximating line segments naturally increases. This is to say that the worst condition at crossover is when both incident and departure angles are at their minimum value of 45°. Logically, this situation necessitates that the Δx interval be halved at crossover. In Figure 21 the best approximation for that interval will be the straight segments that would pass through the crossover point. For analysis, it will be assumed that this approximation will be the closest value of Δt to the actual time for that segment of the trajectory. This is a valid assumption since a



CRS

Figure 21. Maximum Crossover Error Analysis.

better approximation could be made only if the chosen magnitude of Δx were reduced. It follows that the maximum error at crossover can be predicted. The best approximation of Δt between t_1 and t_2 is given by

$$t_b = \frac{\Delta x/2}{\dot{x}_1/2} + \frac{\Delta x/2}{\dot{x}_2/2} = \frac{\Delta x}{\dot{x}_1} + \frac{\Delta x}{\dot{x}_2} \tag{4-1}$$

The computer approximation relates the interval as

$$t_c = \frac{\Delta x}{(\dot{x}_1 + \dot{x}_2)/2} = \frac{2\Delta x}{\dot{x}_1 + \dot{x}_2} \tag{4-2}$$

There exists an error, ϵ, between the best and the computer approximations such that

$$\Delta t_b - \Delta t_c = \epsilon \tag{4-3}$$

It is assumed that $\Delta t_b \geq \Delta t_c$ and $\epsilon \geq 0$; however, the sign of ϵ would justify the relationship. Substituting Equations (4-1) and (4-2) into Equation (4-3) the error is given by

$$(\Delta x/\dot{x}_1 + \Delta x/\dot{x}_2) - 2\Delta x/(\dot{x}_1 + \dot{x}_2) = \epsilon \tag{4-4}$$

Now, for angles of incidence and departure ($\angle_i$ and $\angle_d$) both given equal to 45°, $\dot{x}_1 = \dot{x}_2 = \Delta x/2$. Substitution into Equation (4-4)

$$\Delta x\ (\Delta x + \Delta x)\ /\ \Delta x^2 - 2\Delta x/2\Delta x = \epsilon$$

$$2\Delta x^2\ /\ \Delta x^2 - 1 = \epsilon$$

$$\epsilon = 1 \qquad (4\text{-}5)$$

Therefore, the maximum error that the computer could impose at a crossover would be one unit of the time base, whether it be microseconds, hours, or another unit. The majority of the system trajectories will have crossover angles greater than 45° and the corresponding error would be less than one unit. In the situation where the partition endpoint and crossover coincide, the error is zero. It follows that the deviation from the best approximation is a bounded quantity such that

$$0 \leq \epsilon \leq 1$$

The equivalent percentage error of the entire elapsed time over a trajectory depends on the number of half cycles traced and the general shape of the portrait.

As would be expected, the accuracy of the computer solution is directly dependent upon how well the straight line segments approximate the actual trajectory. The majority of the portraits are segments of arcs. Since in the computing scheme, Δx is fixed throughout the trace, the "fit" of the approximation is visualized most accurate where the curvature is largest. In like manner, the approximation will lose some accuracy as the distance from the trace to the origin decreases. An analysis can be made of this situation with general results.

The time response of a circular trajectory can be calculated exactly. Therefore, this special case will be used as a check on the computer. Figure 22 shows a quadrant of a circular trajectory. The exact elapsed time is independent of the radius and is equal to the traced angle in radians. A scale factor only changes the result by the scale constant. Here, the elapsed time is equal to $\pi/2 = 1.571$ units. Notice that the radius of any trajectory, for the purposes of this discussion, is assumed to be an integer multiple of Δx. This assumption simplifies computations with no loss of generality. Mathematically the computer accepts information according to the following analysis:

$$b_n = (r_n) \sin \alpha_n$$

$$\text{number of } \Delta x\text{'s} = r/\Delta x = N$$

$$\Sigma \Delta t = t = \sum_{n=1}^{n=N} 2\Delta x/(b_{n-1} + b_n) \tag{4-6}$$

If n is known, then

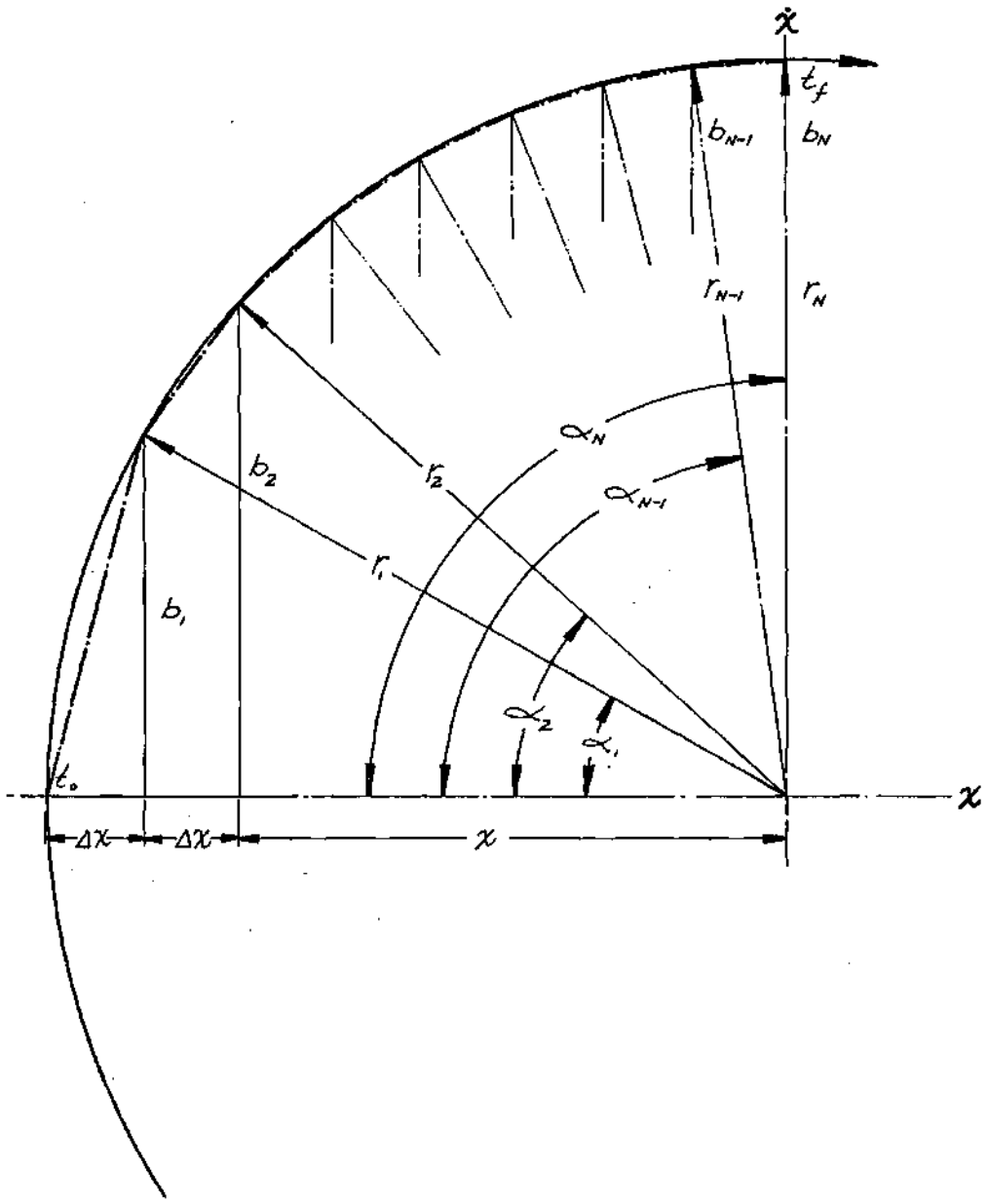
$$x = r - n\ (\Delta x)$$

$$\cos \alpha_n = x/r = \left[r - n\ (\Delta x)\right]/r = 1 - n\ (\Delta x)/r \tag{4-7}$$

$$\alpha_n = \cos^{-1}\left[1 - n\ (\Delta x)/r\right]$$

$$t = 2\Delta x \sum_{n=1}^{n=N} 1/(r \sin \alpha_{n-1} + r \sin \alpha_n) \tag{4-8}$$

Substituting for α_n and α_{n-1} into Equation (4-8)



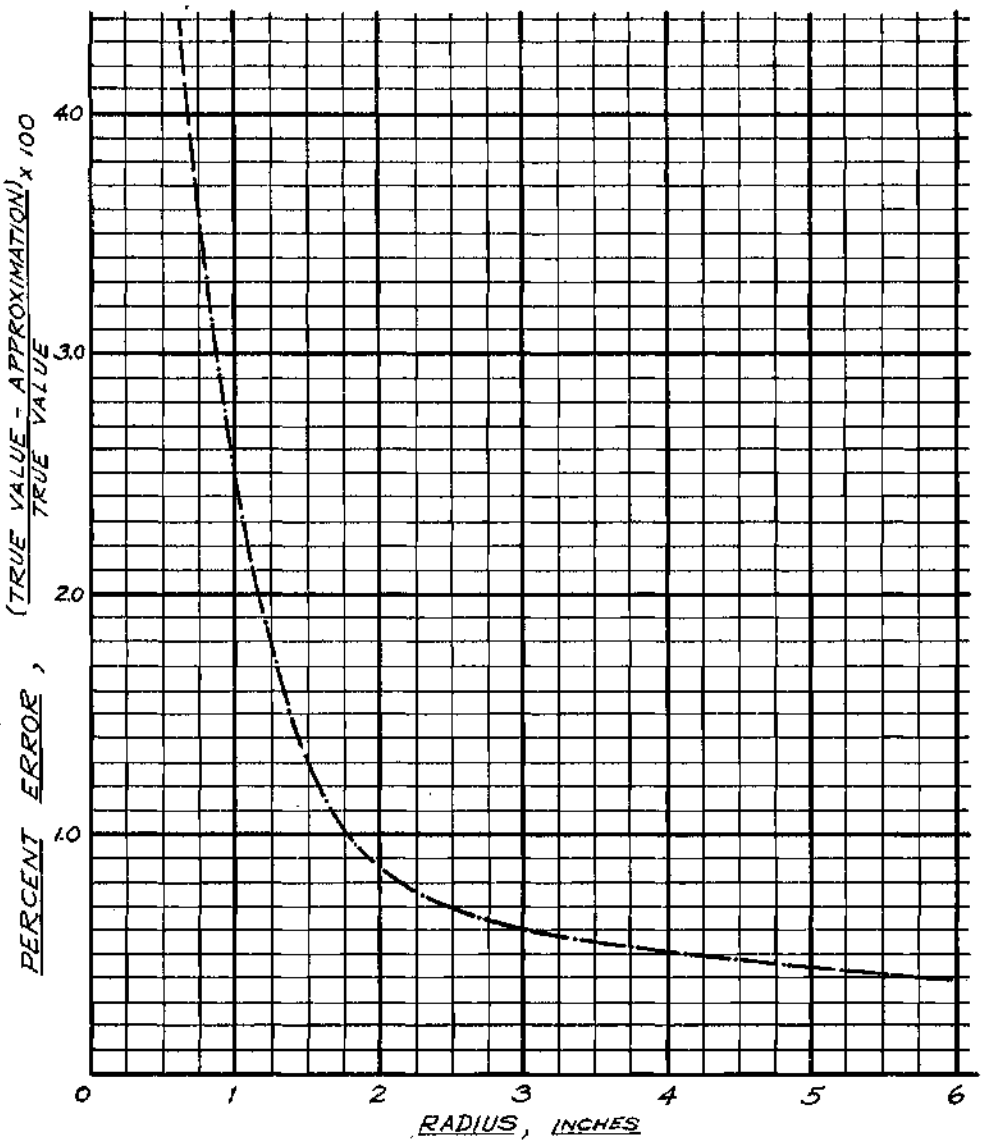
CRS

Figure 22. Segmenting Fit Analysis With Circular Trajectory.

$$t = 2\Delta x/r \sum_{n=1}^{n=N} 1/\left\{\sin(\cos^{-1}[1-(n-1)\Delta x/r]) + \sin(\cos^{-1}[1-n\Delta x/r])\right\} \qquad (4\text{-}9)$$

By actually performing the summation in Equation (4-9) for trajectories of several different radii, it is shown that the deviation of the computer solution from the true answer will increase with a decrease in radius. It is also noted that the approximate solution is always larger than the actual solution. This is to be expected since the $\dot{x}_{avg}$ values in all cases are less than the true $\dot{x}$ magnitudes at corresponding points. The accuracy of the computer technique is quite good at the maximum radius. The graph of Figure 23 shows the relationship of the error as a percentage of the true time plotted versus the radius of the circle. The increase in error with decrease in radius is entirely due to the constant value of Δx. However, most information is taken from the trajectory in the region of larger radius. If a more accurate analysis of the area close to the origin is required, the particular portion of the trajectory can be enlarged by a scale factor and investigated with better results.

The second type of error mentioned is that associated with any deviation from the trajectory line in actually tracing the path. However, the method by which the portrait is partitioned eliminates this error entirely if the stylus traces through each interval endpoint. To the computing mechanism, the trajectory exists only as a continuous line of points, each a projected Δx distance apart. It is remembered that computer summing action is initiated only at partition boundaries;



CRS

Figure 23. Percent Error Versus Radius of Trace.

therefore, no penalty is paid in accuracy if the stylus deviates from the trace between endpoints. The only way that the deviation would affect the solution is if the x projection crosses itself and in effect causes a partitioned interval that is not Δx in length. Figure 24 shows this condition clearly. When the trace crosses itself, it results in an error since the computer sees Δx as only a magnitude of horizontal displacement. The actual solution error is apparent in the $\dot{x}_{avg}$ values. The magnitude of the trace deviation can be related so that a comparison between the trace value and the true value of the velocity is found.

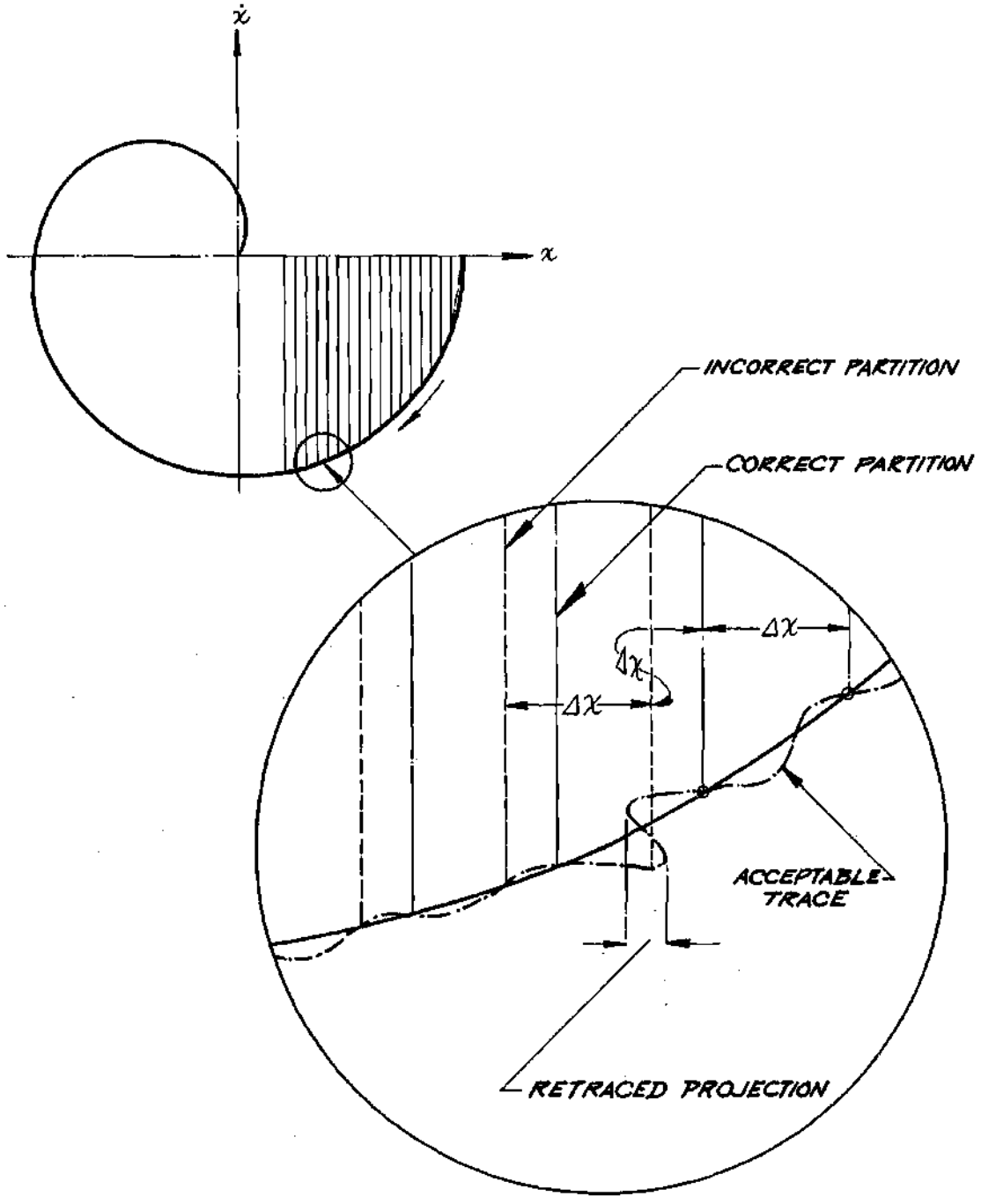
$$\Delta t_{(true)} = \Delta x / \dot{x}_{avg\ (true)}$$

$$\Delta t_{(error)} = \Delta x / \dot{x}_{avg\ (error)}$$

$$\% \text{ error} = \frac{\Delta t\ (true) - \Delta t\ (error)}{t\ (true)} \times 100$$

$$\% \text{ error} = \left[1 - \frac{\dot{x}_{avg\ (true)}}{\dot{x}_{avg\ (error)}}\right] \times 100 \qquad (4\text{-}9)$$

The entire error analysis of this chapter is based on the premise that the graphical trajectory is precisely constructed. In general this is not so. Most portraits are sketched by using isoclines and construction aids such as nodes, foci, saddle points, and other singularities. In some cases, the portrait may include approximations such as backlash, hysteresis, or dead band. The use of the phase plane as a design or analysis tool is not necessarily enhanced by striving for an extremely accurate portrait. The technique is most valuable as a



CRS

Figure 24. Trace Overlap Error Analysis.

method of quickly determining response and overall system activity. It is an indicator of problem areas in design such as component matching. For this reason, the computer fulfills its objective of providing flexible, quick, time response analysis which, in general, would be at least as accurate as the constructed phase trajectory.

APPENDIX

LITERATURE CITED

(1) G. J. Thaler, M. P. Pastel, Analysis and Design of Nonlinear Feedback Control Systems, 82-83

(2) J. E. Shigley, Theory of Machines, 334-336

(3) J. E. Gibson, Franz B. Tuteur, Control System Components, 320-323

(4) J. E. Shigley, Theory of Machines, 334